

DISSERTATION

ROTATION OF CELL SURFACE AND DISSOLVED BIOMOLECULES EXAMINED BY
FLUORESCENCE IMAGING, TIME-TAGGED SINGLE-PHOTON COUNTING, AND
FLUORESCENCE DEPLETION ANISOTROPY

Submitted by

Jason M. Pace

Department of Chemistry

In partial fulfillment of the requirements

For the Degree of Doctor of Philosophy

Colorado State University

Fort Collins, Colorado

Summer 2022

Doctoral Committee:

Advisor: B. George Barisas

Debbie C. Crans

Deborah A. Roess

Alan K. Van Orden

Copyright by Jason Michael Pace 2022

All Rights Reserved

ABSTRACT

ROTATION OF CELL SURFACE AND DISSOLVED BIOMOLECULES EXAMINED BY FLUORESCENCE IMAGING, TIME-TAGGED SINGLE-PHOTON COUNTING, AND FLUORESCENCE DEPLETION ANISOTROPY

In this dissertation, I discuss our studies examining protein rotation both in solution and on single cells. Chapter I gives background on physics of rotational diffusion, the application of these measurements to cellular systems, and a general overview of the field, including a survey of techniques that have been used to measure rotation of membrane proteins. In the next two chapters, I discuss our research on the effect of various cell treatments known to perturb the dynamics of membrane proteins on the rotation of the high-affinity Type I IgE receptor (Fc ϵ RI) expressed on RBL-2H3 cells. I investigated effects on receptor rotation resulting from treatment with IgE antibody as well as from four treatments with IgE and an additional agent including DNP-BSA, paraformaldehyde, M β CD, and cytochalasin D. These agents have varied effects that I expect to cause a significant perturbation of the rotational dynamics of the receptor. These effects range from receptor crosslinking by DNP-BSA and paraformaldehyde which would be expected to hinder receptor rotation to effects on membrane cholesterol content and the underlying cytoskeleton in the cases of M β CD and cytochalasin D, the effects of which are more uncertain and thus of particular interest. I have investigated these phenomena using a single-particle fluorescence imaging approach and, alternatively, a time-tagged single photon counting approach. These topics are the subject of Chapters II and III respectively. These two approaches, while both designed with the intent to investigate the rotational dynamics of membrane proteins

using fluorescence microscopy, share little in common with regards to their methods of data collection and analysis. The concepts behind them are completely different and they use an entirely different set of analysis programs. Chapter IV consists of a published manuscript entitled “Continuous fluorescence depletion anisotropy measurement of protein rotation” which describes our work using a newly-developed pump-probe technique to examine protein rotation in solution and extends this to single-cell measurements. In the continuous variant of fluorescence depletion anisotropy used here, the intensity and polarization of a laser beam are modulated continuously by a programmed acousto-optic modulator and Pockels cell respectively to produce the desired excitation waveform. We have used this method to examine rotation of eosin conjugates of carbonic anhydrase, BSA, and immunoglobulin G in 90% glycerol at varying temperatures. We have also explored the potential application of this method to single-cell measurements and recorded preliminary results on eosin-IgE-bound FcεRI. Generally, we found good agreement with time-resolved phosphorescence anisotropy measurements of rotation of solution-phase molecules and of cell surface FcεRI. Chapter V discusses future avenues worth exploring which would improve upon the methods presented in Chapters II and III. These include faster cameras to access shorter timescales, gold nanorods to improve the signal-to-noise ratio, and a method to obtain a true anisotropy in a microscope.

ACKNOWLEDGEMENTS

I would first like to express my gratitude to my advisor Dr. George Barisas for his mentorship and guidance during my research career here at CSU. Whether it was with experimental work, data analysis, or preparation of this thesis and other manuscripts, his advice and direction has been an indispensable aid. I would also like to thank Dr. Deborah Roess, co-director of our lab and a member of my committee. It was always very helpful to have a second opinion on my writing and her insight into the biological aspects of my projects was very valuable.

I would also like to thank the other members who served on my committee: Dr. Alan Van Orden and Dr. Debbie Crans. I appreciate their generous time, support, and encouragement during this process.

I would like to thank my senior and former lab members who have provided assistance or with whom I have collaborated with at some point or another during my time in the Barisas' lab: Dr. Dongmei Zhang and Dr. Duaa Althumairy. Dr. Zhang and Dr. Peter Winter, a previous member of our lab, made substantial contributions to the experimental portion of this work involving both wet chemistry and the instrumental collection of data.

Additionally, I would like to thank Professor Israel Pecht of the Weizmann Institute of Science, Rehovoth, Israel, for his kind gift of the A2 IgE antibody used in both the imaging and photon-counting studies as well as the preliminary cellular studies using CFDA.

Lastly, I would like to extend a special thanks to my family members who have provided support through this process. Thank you for bearing with me during the stressful periods and lending a listening ear when I encountered difficulties.

DEDICATION

This dissertation is dedicated to my family.

TABLE OF CONTENTS

ABSTRACT.....	ii
ACKNOWLEDGEMENTS.....	iv
DEDICATION.....	vi
Chapter I: Introduction to rotation	1
1.1. Importance of measurements of rotational diffusion of cell membrane proteins	1
1.1.1. Special significance of membrane receptor dynamics	1
1.1.2. Advantage of kinetic approaches over FRET	1
1.1.3. Advantage of measuring rotational diffusion instead of lateral diffusion.....	2
1.1.4. Examples of molecular interactions	5
1.2. Optical techniques to measure cell membrane protein rotation	5
1.2.1. Survey of previously-used optical techniques to measure protein rotation.....	5
1.2.2. Drawbacks to above techniques	7
1.3. References	9
Chapter II: Examination of rotation of quantum dot-labeled type I Fcε receptor by fluorescence imaging	10
2.1. Overview	10
2.2. Introduction	10
2.2.1. Fluorescence imaging approach to visualize QD-labeled receptors	11
2.2.2. Asymmetric nanoparticle labels to probe membrane protein rotation	12
2.2.3. Fluorescence polarization to measure rotation of QD-labeled proteins	20
2.2.4. Type I Fcε Receptor	31

2.2.5. Goals of study.....	36
2.3. Materials and methods	38
2.3.1. “Wet” chemistry and cell culture procedures.....	38
2.3.2. Imaging collection and analysis	40
2.4. Results and Discussion.....	55
2.4.1. Exponential decay of polarization fluctuation TACF is due to rotational re-orientation and not blinking	55
2.4.2. Lateral diffusion does not distort polarization fluctuation TACF.....	56
2.4.3. Reproducibility of Data Collected using Non-Polarized Excitation	64
2.4.4. Comparison of polarized to non-polarized groups	70
2.4.5. Effects of cell treatments on rotation as observed by imaging	71
2.4.6. Comparisons of analysis programs	96
2.4.7. Theoretical expectations for polarization TACF amplitudes	97
2.4.8. Theoretical estimations for rotational correlation times.....	98
2.5. Conclusions	100
2.6. References	102
Chapter III: Examination of rotation of quantum dot-labeled type I Fcε receptor by time-tagged single-photon counting.....	108
3.1. Overview	108
3.2. Introduction	109
3.2.1. Time-tagged single photon counting approach to measure QD-labeled receptor rotation.....	109
3.2.2. Fluorescence correlation for asynchronous time-tagged single-photon counting	

data	111
3.2.3. Goals of study.....	113
3.3. Materials and methods	113
3.3.1. “Wet” chemistry procedures	113
3.3.2. Photon-counting data collection and analysis	114
3.4. Results and Discussion.....	137
3.4.1. Sample data from time-tagged single-photon counting experiments.....	137
3.4.2. Comparing time-tagged single photon counting data for Quantum Dot 605 and 655	147
3.4.3. Effects of cell treatments as observed by time-tagged single-photon counting	154
3.5. Conclusions	168
3.6. References	170
Chapter IV: Continuous fluorescence depletion measurement of protein rotation.....	172
4.1. Overview	172
4.2. Introduction	173
4.3. Theory underlying data analysis	179
4.4. Materials and methods	186
4.4.1. Apparatus and data acquisition	186
4.4.2. Waveform selection.....	188
4.4.3. Data obtained.....	189
4.5. Results and discussion.....	191
4.5.1. Protein rotation vs. temp., viscosity	194
4.5.2. Initial and limiting anisotropy values	198

4.5.3. Lifetime values and distributions	199
4.5.4. Preliminary cellular results	199
4.6. Conclusions	201
4.7. References	204
Chapter V: Overview and future directions	209
5.1. Faster cameras	209
5.2. Gold nanorods	212
5.3. Method to obtain true anisotropy in a microscope	215
5.4. Summary of studies	216
5.5. Significance of studies	218
5.6. Broader importance of work.....	220
5.7. References	224
Appendices.....	225
Program Code #1: Image Correlation Mod15k.....	225
Program Code #2: Image Correlation v.122	282
Program Code #3: Image Correlation v.140	332
Program Code #4: MergeFiles v.39	375
Program Code #5: Photon-Counting Correlation v.80.....	385
List of abbreviations	404

Chapter I: Introduction to rotation

1.1. Importance of measurements of rotational diffusion of cell membrane proteins

1.1.1. Special significance of membrane receptor dynamics

Integral membrane proteins fulfill a variety of cell functions and play important roles in cellular activation, transmembrane transport, and cellular metabolism. Membrane receptors are of particular importance, as they provide a way for exogenous molecules to indirectly relay information to the inside of the cell without actually needing to transit across the membrane. Few molecules possess the requisite characteristics (namely small size and hydrophobicity) to traverse the membrane and so a significant amount of information is transferred from the exterior to the interior of a cell through this indirect route. Consequently, receptors account for a large proportion of drug targets and therefore rightfully command a considerable degree of interest in contemporary cell biology.

1.1.2. Advantage of kinetic approaches over FRET

The standard and currently the most widely-used method for investigating intermolecular interactions between such proteins within the cell membrane is fluorescence resonance energy transfer (FRET) and its associated variants. While these techniques are useful for examining interactions with known species, they are of limited utility when applied to a situation in which one of the binding partners is not known because both species must be labeled and their identities

known in order to conduct such an experiment. Furthermore, these techniques provide little information on the local environment of the protein. In such situations, alternative approaches are required.

1.1.3. Advantage of measuring rotational diffusion instead of lateral diffusion

Kinetic approaches which monitor lateral or rotational diffusion are well-suited to such situations. Both types of measurements reflect the molecular size, shape/conformation, and, by extension, the molecular interactions in which a protein participates. Larger molecules generally exhibit slower diffusion and, by the same principle, molecules undergoing interactions with other molecules exhibit slower diffusion than would be the case if such interactions did not occur. Furthermore, these techniques are sensitive to the local environment and ambient temperature to which the molecule is subjected. Generally, if a molecule is exposed to a more viscous environment, it will experience slower rotation. A decrease in temperature would have a similar effect indirectly through this same mechanism. The dependencies of diffusion on these two factors can be more formalized more rigorously through a number of established mathematical equations. In the case of translational diffusion for a molecule approximated as a sphere, the following relationship, the well-known Stokes-Einstein equation, is observed:

$$D_t = \frac{kT}{6\pi\eta r} \quad (1.1)$$

where D_t is the translational diffusion constant in m^2/s , k is Boltzmann's constant ($1.38 \cdot 10^{-23}$ J/K), T is the absolute temperature in K, η is the viscosity in decapoise (Pa·s or, in more

fundamental units, $\text{kg}\cdot\text{m}^{-1}\cdot\text{s}^{-1}$), and r is the hydrodynamic radius in m. The rotational diffusion coefficient, D_r , can similarly be defined as:

$$D_r = \frac{kT}{8\pi\eta r^3} \quad (1.2)$$

This is also sometimes written in terms of the hydrated volume V_{hyd} as:

$$D_r = \frac{kT}{6\eta V_{\text{hyd}}} \quad (1.3)$$

where $V_{\text{hyd}}=4\pi r^3/3$ [2, 3]. The rotational correlation time is inversely related to the rotational diffusion constant as $\Phi=1/6D_r$ and is therefore directly proportional to the hydrated specific volume as:

$$\phi = \frac{\eta V_{\text{hyd}}}{kT} \quad (1.4)$$

Thus, in three-dimensional systems, the rotational diffusion constant differs from the translational diffusion constant by a factor of $3r^2/4$ and is therefore much more sensitive to molecular size. In fact, the difference in the sensitivity is even more pronounced than this because, for lateral diffusion in two dimensions as is the case for a receptor embedded in the membrane (modeled as a cylindrical inclusion), the Stokes-Einstein equation is no longer applicable and the translational/lateral diffusion constant D_t exhibits an even weaker dependence on molecular size, varying inversely with the logarithm of the particle radius as described by the Saffman-Delbrück equation [2] shown below:

$$D_t = \frac{kT}{4\pi\eta_m h} [\ln(\frac{2L_{sd}}{r}) - \gamma] \quad (1.5)$$

where η_m is the membrane viscosity, h is the membrane thickness, L_{sd} is the Saffman-Delbrück length which is equal to $h\eta_m/2\eta_f$ where η_f is the viscosity of the bulk fluid surrounding the membrane, and γ is the Euler-Mascheroni constant, an irrational number with a value of ~ 0.577 . The Saffman-Delbrück also makes a small amendment to the equation for the rotational diffusion coefficient of a membrane protein modeled as a cylindrical inclusion [2]:

$$D_r = \frac{k_B T}{4\pi\eta_m r^2 h} \quad (1.6)$$

Consider the ratio of the value of the translational diffusion constant to that of the rotational diffusion constant using the Stokes- Einstein treatment for isotropic diffusion in 3-D and the Saffman-Delbrück treatment for diffusion in 2-D. In the 2-D membrane case, this ratio is increased by a factor of ~ 4 . This is not surprising when considering how the velocity fields decline in the 3-D and the 2-D case for rotational relative to lateral diffusion. For rotation, the velocity fields decline by $1/r^2$ in 3-D compared to $1/r$ in 2-D whereas, for lateral diffusion, the sensitivity to molecular radius is weaker, decaying as $1/r$ in 3-D and $-\log(r)$ in 2-D. In both cases, the sensitivity of the diffusion constant to molecular radius is lowered in a membrane. This is because, while energy can be dissipated in 3-D, in 2-D this is not the case and flow fields representing a membrane particle's motion extend into the aqueous layer quite a distance. The decline in sensitivity is more drastic for translation than it is rotation. For rotation, the energy

dissipated is localized and the receptor's resistance to rotation is limited practically completely by the viscous membrane and the less viscous aqueous layer makes little contribution. For translation, energy is dissipated over a larger area and the translational drag declines because of low dissipation of energy into the aqueous phases. Given the greater sensitivity (especially in the 2-D case as in a membrane), measurement of rotational diffusion is a better candidate than lateral diffusion to elucidate processes such as receptor oligomerization and aggregation.

1.1.4. Examples of molecular interactions

Consider some common events known to occur within the context of the cell membrane which can be examined via measurement of protein rotation and can be broadly classified under the aforementioned categories of receptor interactions or changes in the local environment of the receptor. Examples of the former would include receptor interactions with the cytoskeleton, receptor binding to ligand, or crosslinking of sensitized immune receptors with polyvalent antigen, hapten-carrier adducts, or fixative agents. An example of the latter would be confinement of receptor to compartments which possess different local compositions and viscosities than the surrounding bulk membrane.

1.2. Optical techniques to measure cell membrane protein rotation

1.2.1. Survey of previously-used optical techniques to measure rotation

For the investigator who wishes to examine rotation of cell membrane proteins, a variety of optical techniques are available. These techniques may measure emission, absorption, or scattering. A commonality between these techniques is their reliance on the measurement of anisotropy or polarization, quantities that reflect the orientational asymmetry in the light distribution resulting from the rotational motion of a molecule. A variety of techniques have been used to measure the rotation of ensembles of molecules on both single cells and in cell suspensions. *Time-resolved fluorescence anisotropy* has been used to measure the rotation of fluorescent lipid probes in membranes on timescales on the order of several nanoseconds. However, the inherent transience of fluorescence lifetimes imposes an upper bound on the rotational correlation times that can be measured with such a technique. Unfortunately, given that the rotational diffusion of individual, unhindered cell membrane proteins is expected to occur on the 10-100 μ s timescale, the technique cannot access timescales long enough to be applicable to such systems. Other techniques which rely either directly or indirectly on the kinetically slower phenomena of triplet decay and, in some cases, measurement of the associated phosphorescence lifetimes are better suited for measurements of membrane protein rotation. Amongst these techniques are *linear dichroism* [4], *E-type fluorescence* (also known as delayed fluorescence) [5], *time-resolved phosphorescence anisotropy* [6], and *fluorescence depletion anisotropy* (also known as polarized fluorescence depletion) [7, 8]. Of these techniques, time-resolved phosphorescence anisotropy and E-type fluorescence have enjoyed the greatest popularity, largely owing to their relative simplicity and ease of use. However, while time-resolved phosphorescence anisotropy enables measurement of the microsecond-timescale rotational correlation times associated with membrane protein rotation, a disadvantage stemming from this method's reliance on phosphorescence emission is its low sensitivity. Compared to time-resolved

phosphorescence anisotropy, FDA possesses greater sensitivity because the signal actually monitored by this method is fluorescence which has a much greater photon flux rate than phosphorescence. Thus FDA is unique in that it involves triplet decay yet measures fluorescence and thus has both the capability to measure the longer rotational correlation times associated with membrane proteins and the sensitivity conferred by the higher photon throughput of fluorescence emission. In FDA, a low-intensity laser beam is used to excite chromophores which then emit steady-state fluorescence. A short-duration, high-intensity laser beam is then used to induce a fraction of chromophores to undergo intersystem crossing to a long-lived triplet state, producing a decay in the fluorescence signal. The gradual return to the singlet ground state replenishes the fluorescence signal. Rotational reorientation also makes a contribution to the fluorescence recovery and thus the superimposition of these two phenomena produces the recovery observed. Analysis of such a decay involves parsing out the contribution from each and extracting the rotational information contained within. Multiple variants of FDA exist, including time- and frequency- domain variants. We have developed a continuous variant of FDA (CFDA) which combines the merits of time- and frequency- domain variants into a single approach. These studies have resulted in a publication in the *Journal of Fluorescence* [1] and are also presented in this thesis as the subject of Chapter IV.

1.2.2. Drawbacks of above techniques

However, all of these techniques suffer from drawbacks. One issue is that these techniques are designed for ensemble measurements of protein rotation. Consequently, while they may provide information on the average properties of a molecule, they offer no information

on individual molecules. This is problematic because receptors may exhibit varying degrees of aggregation giving a distribution of sizes. Additionally, they may experience drastically different environments from one molecule to the next (*e.g.* a receptor confined to a lipid raft versus one experiencing the lower viscosity of the bulk membrane). Another limitation of these techniques is their inability to access even longer timescales, a matter which our imaging technique, the subject of Chapter II, is able to address.

1.3. References

1. Zhang, D., et al., *Continuous Fluorescence Depletion Anisotropy Measurement of Protein Rotation*. Journal of Fluorescence, 2018. **28**: p. 533-542.
2. Saffman, P. and M. Delbruck, *Brownian motion in biological membranes*. Proc Natl Acad Sci, 1975. **72**(8): p. 3111-3113.
3. Fooksman, D., M. Edidin, and B. Barisas, *Measuring rotational diffusion of MHC class I on live cells by polarized FPR*. Biophys Chem, 2007. **130**(1-2): p. 10-16.
4. Cone, R., *Rotational diffusion of rhodopsin in the visual receptor membrane*. Nat New Biol, 1972. **236**: p. 39-43.
5. Greinert, R., et al., *E-type delayed fluorescence depolarization, technique to probe rotational motion in the microsecond range*. J Biochem Biophys Methods, 1979. **1**(2): p. 77-83.
6. Austin, R., S. Chan, and T. Jovin, *Rotational diffusion of cell surface components by time-resolved phosphorescence anisotropy*. Proc Natl Acad Sci USA, 1979. **76**(11): p. 5650-5654.
7. Johnson, P. and P. Garland, *Depolarization of fluorescence depletion: A microscopic method for measuring rotational diffusion of membrane proteins on the surface of a single cell*. FEBS Letters, 1981. **132**(2): p. 252-256.
8. Johnson, P. and P. Garland, *Fluorescent triplet probes for measuring the rotational diffusion of membrane proteins*. Biochem J, 1982. **203**: p. 313-321.

Chapter II: Examination of rotation of quantum dot-labeled type I Fcε receptor by fluorescence imaging

2.1. Overview

In this chapter, we describe our studies employing fluorescence imaging to examine rotation of the type I Fcε receptor. This imaging approach involves correlation analysis of orientational fluctuations of FcεRI conjugated to a fluorescent quantum dot probe. Measurement of these orientational fluctuations was achieved by imaging a single quantum dot-labeled receptor at multiple discrete points in time and monitoring the progressive change in fluorescence polarization upon rotation. This technique enables examination of receptor rotational motion on a timescale ranging from tens to hundreds of milliseconds—a timescale on which there has been little previous study. We investigated the effect of various cell treatments expected to alter receptor rotation. We observed rotational correlation times scattered within a range of ~ 10 - 10^3 ms amongst individual copies of the receptor with weighted geometric mean rotational correlation times within the ~ 100 - 150 ms range. As such times exceed that which could be expected to reflect the hydrodynamic rotation of the receptor, we interpreted this to reflect the rotational or librational motion of localized domains containing these receptors and/or, depending on treatment, interactions of the receptor with other proteins which significantly slow its rotation.

2.2. Introduction

2.2.1. Fluorescence Imaging Approach to Visualize QD-Labeled Receptors

The advent of asymmetric, rod-shaped nanoparticles conjugatable to certain membrane proteins, including quantum dots (QDs) used in this study and gold nanorods, enables single-molecule examination of rotation occurring on a time regime inaccessible to the techniques described in the previous chapter. Although time-resolved phosphorescence anisotropy is useful for examining hydrodynamic rotation of receptors occurring on the aforementioned 10-100 μ s timescale, it does not provide information on timescales significantly longer than this. In time-resolved phosphorescence anisotropy measurements on these proteins, the phosphorescence lifetime of the probe used places an upper bound on the timescale that can be examined. An example of the magnitudes of phosphorescence lifetimes typically encountered is that of the commonly-employed phosphor erythrosin, which has a lifetime of about 500 μ s in deoxygenated samples. Thus these measurements are generally limited to measurement of rotational motion occurring on a timescale no longer than about 5 ms. What is observed with this technique is a limiting/residual anisotropy which reflects a rotationally immobile fraction of receptors which could possibly decay on a longer timescale. Rotational motion occurring on a longer timescale is accessible to polarized photobleaching measurements; however, such measurements require a minimum of 100 ms to bleach fluorophores. Consequently, there has been little study on the rotational motion of membrane receptors with correlation times between about 5 ms to 100 ms. To examine this timescale, we have employed a fluorescence correlation imaging technique. Specifically, we have applied this technique to examine 1-1000 points at 10 ms per point giving an accessible time range of about 10-10,000 ms (0.01-10 s). The lower constraint on this range is determined by the frame rate that can be recorded by our camera. As for the upper bound, though

theoretically our technique has an acquisition time of 10 s, in practice not much useful correlation information is found for correlation times exceeding 1000 ms. Thus, while this technique is expected to provide little information on the timescale for hydrodynamic rotation of membrane proteins (~10-100 μ s), an advantage of the technique is its ability of probe rotation on a longer timescale which may be characteristic of hindered protein rotation or rotation of domains to which the receptor is confined.

2.2.2. Asymmetric Nanoparticle Labels to Probe Membrane Protein Rotation

Quantum dots (QDs) are semiconductor nanoparticles with dimensions on the order of a few nanometers. When these nanoparticles absorb light, they undergo a transition in which an electron within the QD is excited from the valence band to the conductance band. When this electron relaxes to a lower energy state and returns to the valence band, light is emitted. QDs can be composed of a variety of different materials including PbS, PbSe, CdS, CdTe, InAs, and InP, amongst others. Our QDs consist of a cadmium selenide (CdSe) core and a zinc sulfide (ZnS) shell. The ZnS shell is coated with a third layer which is chemically inert thereby preventing non-specific interactions. This coat can be composed of various materials including polymers, peptides, or amphiphilic micelles. The coat can be conjugated to biomolecules such as the protein streptavidin which has a high affinity for biotin and can thereby be strongly linked to biotinylated proteins. A schematic of a QD showing the various layers and their compositions and molecules they can be conjugated to is depicted in Figure 2.1.

Depending on their aspect ratio, CdSe nanoparticles can be considered to possess a one-

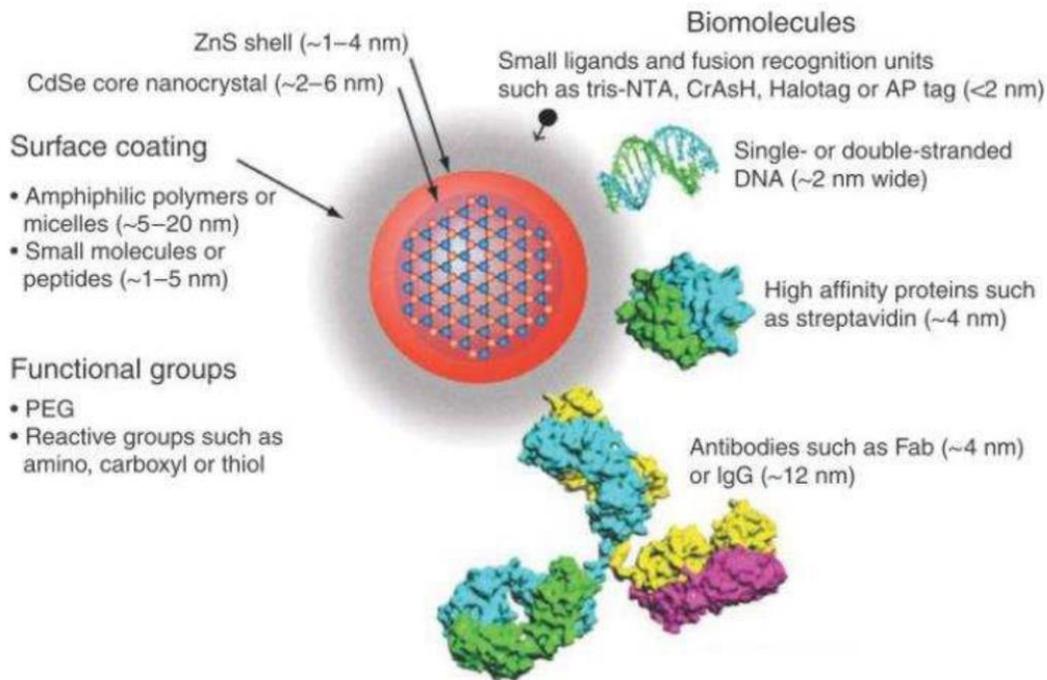


Figure 2.1: Schematic diagram of QD with a CdSe core and ZnS shell. The ZnS shell can be coated with a variety of materials including amphiphilic polymers, amphiphilic micelles, small molecules, or peptides. Functional groups such as polyethylene glycol (PEG) or amino, carboxyl, and thiol groups allow for the conjugation of various biomolecules to the QDs to form a QD-biomolecule adduct. In our studies, we have conjugated the QD to streptavidin, a protein with high affinity for biotin. This formed the linker to biotinylated DNP-specific A2 IgE antibody. Image adapted from Pinaud *et al.*, 2010 [1].

or a degenerate two-dimensional absorption dipole [2]. If the aspect ratio is greater than two, then the QD behaves as a one-dimensional absorber with a single transition absorption dipole oriented along the principal C3 axis. If the aspect ratio is less than two, our QDs behave as two-dimensional absorbers. For such an absorber, light can be absorbed polarized in either of two orthogonal orientations within the equatorial plane of the molecule. This plane is known as the “bright plane” of the molecule. Light polarized along the long C3 principal axis of the molecule, on the other hand, is incapable of exciting the molecule and this is hence appropriately referred to as the “dark axis” of the molecule. Figure 2.2 shows the geometries and orientations of the absorption transition dipoles for one- and two-dimensional absorbers.

One potential concern surrounding the use of nanoparticle labels to probe membrane protein rotational motion is the possibility that the nanoparticles themselves could perturb this motion, thereby altering the very characteristic of the system they are intended to enable us to measure and rendering them unsuitable for this application. These types of concerns fall into two categories: concerns that conjugation of quantum dots to the protein could itself inhibit rotation of the protein, and concerns that flexibility of the linker to the quantum dot probe could degrade measurements of protein rotation.

Fortunately, theoretical predictions suggest that conjugation to these nanoparticles will not encumber protein rotation to an appreciable extent [4]. According to theory, molecules with larger friction coefficients will exhibit longer rotational correlation times. In the case of membrane protein attached to a nanoparticle label, this friction coefficient is approximately equal to the sum of the friction coefficients of the protein and the attached label (these are in turn

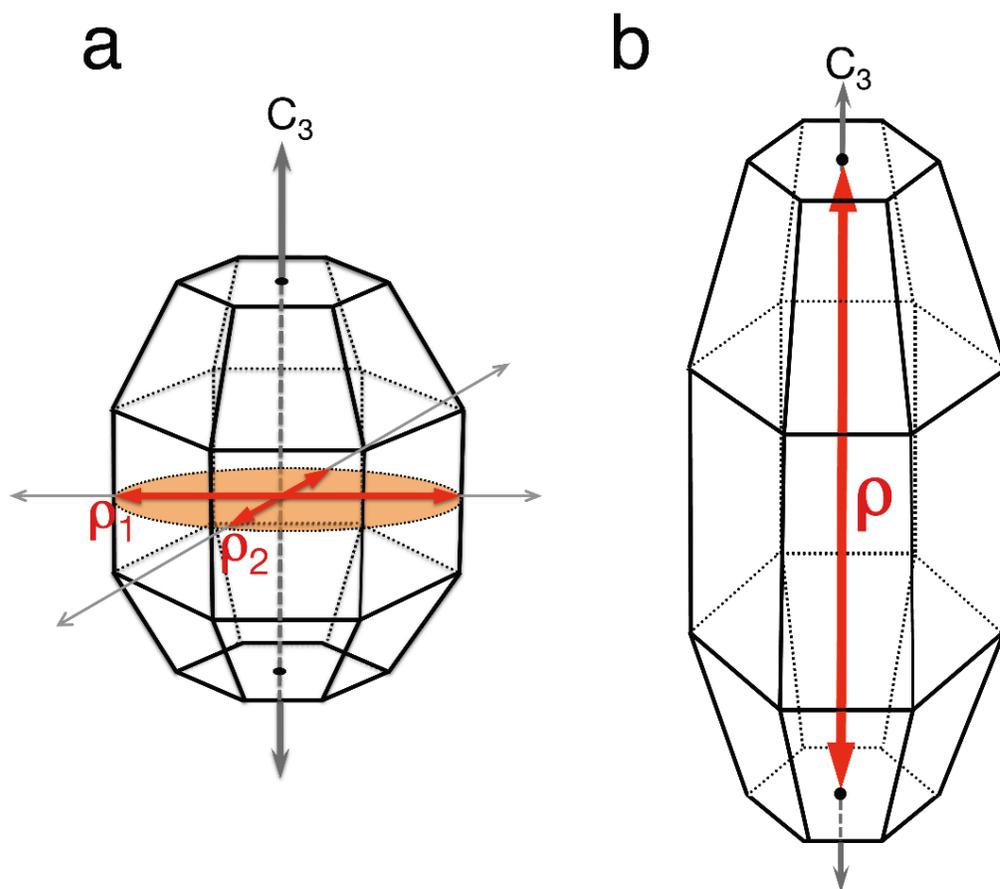


Figure 2.2: Structures and spatial arrangement of absorption transition dipole vectors for CdSe QDs possessing a wurtzite crystal structure. These QDs belong to the C_{3V} point group and their C_3 principal axis runs longitudinally through the QD. Panel A shows a QD with an aspect ratio ≤ 2 . This type of QD possesses a two-dimensional absorption transition dipole which can be thought of as a composite of two orthogonal one-dimensional absorption transition dipoles (ρ_1 and ρ_2) contained within the equatorial plane of the molecule. Panel B shows an elongated CdSe QD with an aspect ratio ≥ 2 . These QDs are considered nanorods and possess a single, one-dimensional absorption transition dipole which is aligned with the principal C_3 axis. Image adapted from Shapiro, 2009 [3].

functions of both their sizes and the local environment to which they are exposed such that an increase in either the size or viscosity of the local environment will result in an increase in the friction coefficient of the respective molecule). As a consequence, the rotational correlation time of the labeled protein is equal to the product of the rotational correlation times of the protein and the nanoparticle label divided by the sum of these correlation times such that an increase in the size and, by extension, rotational correlation time of either the protein or the probe will result in a labeled protein with a longer correlation time. Sizes of Invitrogen Quantum Dots 605 and 655 were estimated from TEM electron micrographs to be 5.1 nm x 10.9 nm and 5.8 nm x 12.8 nm respectively (see Fig. 2.3) [5]. These are similar to values reported by the manufacturer, which were reported to be 5 nm x 12 nm and 8 nm x 15 nm respectively [6]. The longer rotational correlation time of the larger QD655 probe, corresponding to rotation around either of its degenerate short axes, is predicted to be $\sim 0.27 \mu\text{s}$ [7]. This value is for QD655 immersed in aqueous solution which is expected to closely resemble that of a QD attached to a membrane protein because, unlike the membrane proteins they label which are confined to the plane of membrane, the QD probes protrude into the extracellular fluid and will thus exhibit a shorter rotational correlation time than they would if they too experienced the high viscosity of the membrane. If, considering the conservative case in which this larger QD probe is conjugated to small membrane proteins such as MHC class I and class II antigens having rotational correlation times ranging from 25-30 μs , this would give a correlation time which is not substantially different than the rotational correlation time of the unlabeled protein. Even in the case in which these relatively small proteins are labeled with the largest of nanoparticle probes, gold nanorods which dimensions have been measured by TEM as 9.5x35 nm and correlation time 3.5 μs , the rotational correlation time would still only be increased by a negligible amount. To summarize,

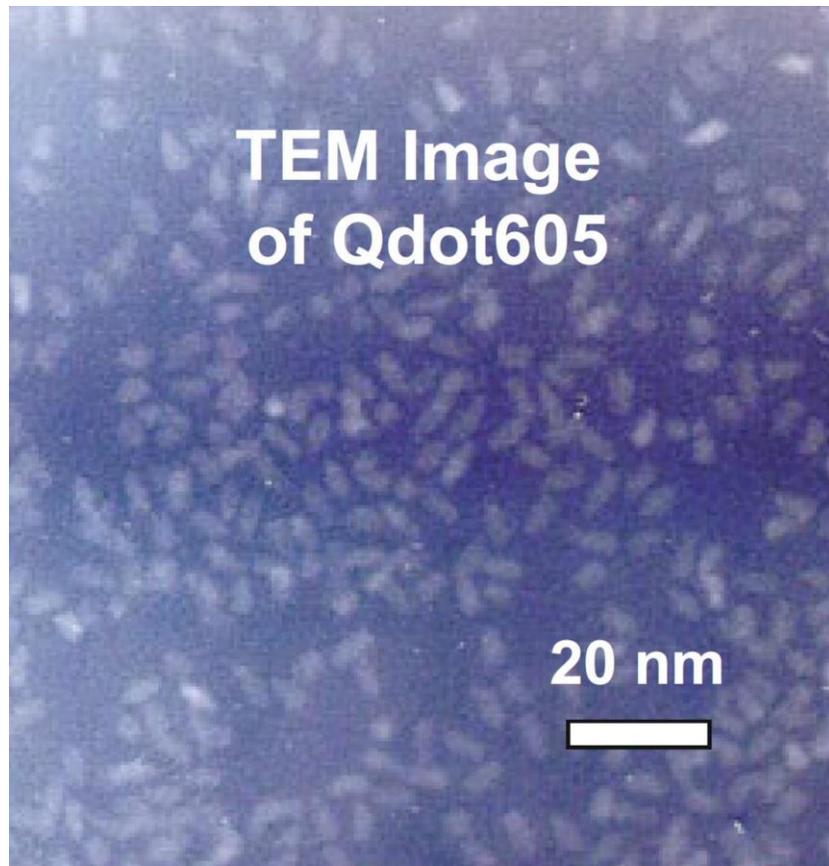


Figure 2.3: TEM image of QD605. Average dimensions were found to be 5.1x10.9 nm. Image adapted from Zhang *et al.*, 2017 [5]

the IgE-bound QD is unlikely to have a significant effect on the rotational correlation time of the receptor because, unlike the receptor, it is exposed to the aqueous extracellular solution and thus experiences a low viscosity whereas the receptor is embedded in the membrane and thus experiences a much higher viscosity. Therefore practically all of the rotational restraint is a consequence of the membrane-embedded receptor's resistance to rotation.

Theoretical considerations also allay concerns that probe flexibility associated with the nanoparticle probes might degrade protein rotation measurements [4]. Though it is true that the nanoparticle is conjugated to the protein through a flexible linker, there will still some restriction in the number of conformational states it can assume by virtue of this linkage. This is because the QD is anchored by this linkage conferring a degree of rigidity and preventing the QD from moving through this bond. For this reason, the anisotropy decay will always contain a contribution corresponding to the rotation of the entire nanoparticle-labeled protein. This mechanism has been shown to account for almost a quarter of the observed anisotropy [8].

An additional concern presented by the use of QDs is their propensity to spontaneously switch between a fluorescent “on” state and a dark “off” state, a phenomenon termed “blinking” [9]. This tendency to fluctuate between these two states is an intrinsic property of QDs. This dark state can persist from nanoseconds up to hundreds of seconds [10, 11]. The probability distributions for these two states are each dictated by their own individual power laws. The probability of observing a given “on” or “off” time is related to the duration of that time in the following fashion:

$$P(\tau) \propto \tau^{-\mu} \quad (2.1)$$

where the exponent μ ranges from 1.1-2.2. For blinking “on” times, μ varies with sample temperature and excitation intensity, but blinking “off” times are independent of these factors [12]. As will be discussed in depth later on, QD blinking represents a major concern for our experiments because changes in the fluorescence intensity function can manifest to some degree in the polarization function, thereby distorting the polarization TACF. Thus one important aim of data analysis was to develop an optimization procedure in which instrumental parameters were selected so that the statistical dependence of the polarization TACF on the intensity TACF was minimized.

Multiple mechanisms have been proposed to explain the fluorescence intermittency exhibited by QDs. Chronologically, the first of these to be proposed was the Auger ionization model. This empirical model was proposed by Efros and Rosen in a seminal 1997 article in which they analogize QD blinking to a random telegraph signal [13]. According to this model, the cycling of QDs between a radiative “on” state and a non-radiative “off” state can be described as follows. An initially non-emitting QD in the “off” state is excited by low-intensity continuous-wave light. This ejects an electron generating an electron-hole pair which undergoes radiative recombination causing the QD to transition to an “on” state which emits light. This continues until the QD undergoes either thermal or Auger auto-ionization, transitioning back to an “off” state where fluorescence is quenched by a fast non-radiative Auger recombination process. The duration of the “on” period depends on the rate of auto-ionization and exhibits a strong dependence on the excitation intensity. The electron that was ejected creating a “hole” is

temporarily trapped and resides in the surrounding matrix during the “off” phase until it eventually returns to the nanocrystal and the “on” state thereby resumes.

Frantusov and colleagues proposed a phenomenological theory of multiple recombination centers [14] which considers the Auger ionization model for QD blinking inadequate because it fails to account for experimental evidence that suggests that QDs do not exhibit two discrete “on” and “off” states but rather exhibit a spectrum of emission intensities. This model makes an amendment to the Auger ionization model by proposing that non-radiative relaxation occurs via multiple recombination centers. Electron holes are trapped by these quenching centers and undergo non-radiative recombination with the electron. Each recombination center can exist in quasi-stationary active and inactive conformations which differ in trapping ability.

Most recently, the atomistic model was proposed as a mechanism to explain the blinking phenomenon which describes it in terms of surface vacancies [15]. This model suggests that such surface vacancies enhance fluorescence quantum yield, but aggregation of such vacancies actually has the opposite effect and can convert the QD to an “off” state. This model also predicts that foreign cations can stabilize these vacancies, thereby improving quantum yield and has informed approaches to surface passivation of QDs.

2.2.3. Fluorescence Polarization to Measure Rotation of QD-labeled Proteins

Analysis of fluorescence fluctuations due to rotational reorientation yields information on QD rotation. This can be viewed in a manner analogous to the measurement of fluctuations in

particle position as is done in single-particle tracking. Consider an optical configuration in which the optical/longitudinal axis along which excitation occurs is designated as the z-axis, and exciting light can be polarized laterally in a horizontal direction along the x-axis, in a vertical direction along the y-axis, or with some component of each (*i.e.* in any orientation within a space defined by the xy-plane). Conducting an experiment using non-polarized excitation is, in essence, equivalent to what would be obtained if the sample was excited by light polarized in the x- and y-orientations *separately*. It is necessary to emphasize here that carrying out an experiment with non-polarized excitation is *not* equivalent to conducting one with simultaneous x- and y-polarized excitation or, alternatively, every intermediate orientation between these two axes specified by a vector lying in the I/III quadrants in the Cartesian xy-plane. In either case, a vector sum of these vectors would result in a net field vector with equal x- and y- components such that it would amount to polarized excitation at a 45 degree angle between these two axes. This is clearly distinct from what is meant when discussing non-polarized excitation.

Regardless of the polarization of the exciting light, the emission anisotropy, which describes rotational reorientation in three dimensions, is defined in the same manner:

$$r(t) = \frac{I_x - I_y}{I_x + I_y + I_z} \quad (2.2)$$

The denominator of this expression, equal to the sum of the intensities, is also referred to as the intensity function $s(t)$. However, the optical configuration/geometry of our experimental setup imposes limitations on which quantities can practically be measured. Z-polarized emission can only be measured by a T-format fluorometer and typically not in a conventional microscope

as we have used in our experiments. On this note, it is worth mentioning that we have developed a novel method for obtaining z-polarized emission in a microscope. A description of this method is given in future directions, the subject of Chapter Five in this thesis.

Without information on the z-polarized intensity, what is typically measured as a proxy for the “true” anisotropy described by Equation 1.7 is an apparent anisotropy where it is assumed that $I_z=I_y$. This assumption regards these two emission orientations to be equivalent because they are both perpendicular to the orientation of the exciting light and it can thus be rationalized statistically under certain conditions (namely in a situation where we have an ensemble average of many freely-rotating particles). Then we have instead:

$$r(t) = \frac{I_x - I_y}{I_x + 2I_y} \quad (2.3)$$

Or, if excitation is polarized in the x-plane, we can use the alternative notation where I_{\parallel} and I_{\perp} are emission light intensities polarized parallel (\parallel) and perpendicular (\perp) to the polarization of the exciting light:

$$r(t) = \frac{I_{\parallel}(t) - I_{\perp}(t)}{I_{\parallel}(t) + 2I_{\perp}(t)} \quad (2.4)$$

In the experiments described herein, we have chosen to measure the 2-dimensional analogue of the anisotropy known as the polarization which is defined as follows:

$$p(t) = \frac{I_{\parallel}(t) - I_{\perp}(t)}{I_{\parallel}(t) + I_{\perp}(t)} \quad (2.5)$$

Though we could approximate the three-dimensional anisotropy function by assuming the two perpendicular emission orientations (z and y using our current convention) are equivalent as described above, we have instead chosen to measure this polarization function to circumvent this assumption entirely. As such, the “2” drops out of the formula above. Therefore, from this point onwards, we have used polarization. In any case, it is worth noting that both functions contain the same information on rotational reorientation, and can be interconverted using the following equation:

$$p = \frac{3r}{2+r} \quad (2.6)$$

Modeling fluorescence polarization for a QD-labeled protein is complicated by a number of aforementioned factors including the experimental configuration whether the excitation is polarized or non-polarized, and whether the molecule is a one- or two-dimensional absorber. Another consideration is how the single-molecule measurements we have conducted here on cells relate to ensemble measurements in solution or in a suspension. To start, it is useful to consider a simplified case in which a one-dimensional absorber with a single absorption transition dipole in some orientation prior to rotation evolves to some new orientation upon rotation. If the vector corresponding to this absorption transition dipole in some initial orientation is designated \mathbf{A} and has components a_x , a_y , and a_z , then rotation through the internal polar angle θ

(the angle from the z-axis), the internal azimuthal angle ϕ (the angle from the azimuthal x-axis), and the angle corresponding to rotation around the molecular internal z-axis ψ results in an absorption transition dipole in some final orientation described by vector \mathbf{A}' with components a_x' , a_y' , and a_z' . The components of this new vector \mathbf{A}' are given by the matrix product of vector \mathbf{A} and an Euler rotation matrix \mathbf{R} as follows:

$$\mathbf{A}' = \mathbf{R} \cdot \mathbf{A} \quad (2.7)$$

where the Euler rotation matrix \mathbf{R} , a function of the angles θ , ϕ , and ψ , is defined as:

$$R(\theta, \phi, \psi) = \begin{pmatrix} \cos \theta \cos \phi \cos \psi - \sin \phi \sin \psi & -\sin \phi \cos \psi - \cos \theta \cos \phi \sin \psi & \sin \theta \cos \phi \\ \cos \theta \sin \phi \cos \psi + \cos \phi \sin \psi & \cos \phi \cos \psi - \cos \theta \sin \phi \sin \psi & \sin \theta \sin \phi \\ -\sin \theta \cos \psi & \sin \theta \sin \psi & \cos \theta \end{pmatrix} \quad (2.8)$$

If we express vectors in terms of their constituent components and substitute this definition of \mathbf{R} into Equation 2.6, we have the following equation for the components of the new vector \mathbf{A}' in terms of its Cartesian components a_x' , a_y' , and a_z' :

$$\begin{pmatrix} a_x' \\ a_y' \\ a_z' \end{pmatrix} = R(\theta, \phi, \psi) \cdot \begin{pmatrix} a_x \\ a_y \\ a_z \end{pmatrix} \quad (2.9)$$

Thus the matrix product of the 3x3 Euler rotation matrix and the 3x1 initial absorption transition dipole vector gives a 3x1 vector describing the final orientation of the absorption transition

dipole after rotation.

With our expression for the rotated absorption vector, it is possible to determine the orientation and magnitude of the expected fluorescence emission. Given our definition of this rotated absorption vector, the extent of excitation \mathbf{E}' can be defined as the dot (or scalar) product of this vector and the excitation polarization vector \mathbf{E} as follows:

$$\mathbf{E}' = \mathbf{A}' \cdot \mathbf{E} \quad (2.10)$$

The orientation of the resultant fluorescence emission can then be represented by a fluorescence emission field vector \mathbf{F} which is defined as the normal product of the scalar extent of excitation and the rotated absorption transition dipole vector divided by the magnitude of the initial unrotated absorption transition dipole vector:

$$\mathbf{F} = \frac{\mathbf{E}' \mathbf{A}'}{|\mathbf{A}|} \quad (2.11)$$

The accompanying magnitude of the fluorescence intensity \mathbf{I} is defined as the square root of the matrix product of the complex conjugate of the fluorescence emission field vector and the fluorescence emission field vector:

$$\mathbf{I} = \sqrt{\mathbf{F}^* \cdot \mathbf{F}} \quad (2.12)$$

\mathbf{I} is thus a vector with components I_x , I_y , and I_z . By this convention, a one-dimensional absorber with an absorption coefficient of unity and an absorption transition dipole that is in alignment with polarized excitation with an intensity of one would produce an instantaneous fluorescence intensity of one.

With these basic fundamentals established in our model of fluorescence polarization, it is now possible to consider some situations which introduce an additional degree of complexity to our discussion and which are all relevant to our experiments. As previously mentioned, these factors include the experimental geometry used, the nature of the excitation, whether the QD is considered as a one or two-dimensional absorber, and the relationship between the single-molecule experiments we have conducted to ensemble measurements. Regarding the experimental geometry, we had the capability of measuring either an apparent anisotropy or a polarization. As will be discussed, measuring polarization instead of apparent anisotropy provides some distinct advantages. Regarding the QD absorption behavior, there is some debate about whether our quantum dots behave as one- or two- dimensional absorbers, and so it is worthwhile to consider a scenario in which a molecule has more than one absorption transition dipole. In the former case regarding two- dimensional absorbers, one can consider each absorption transition dipole vector and excitation polarization vector combination separately and subsequently add the fluorescence intensity vectors produced in each case. Since we are also interested in what difference there may be (if any) between experiments conducted with polarized and non-polarized excitation, it is additionally worth considering the treatment of the non-polarized case. For a situation involving non-polarized excitation, a similar treatment would be required in which the x- and y-polarized excitation vectors are considered separately and the

fluorescence intensity vectors generated from the combination of each with the absorption transition dipole vector(s) are summed.

Table 2.1 provides a summary of the various experimental parameters that can be expected to result from various combinations of the experimental conditions outlined above (namely the experimental geometry, type of excitation, QD symmetry, and whether the experiment is a single-molecule or ensemble measurement). These parameters include the average intensity \bar{s} and anisotropy/polarization (\bar{r} or \bar{p}) in an ensemble measurement and the corresponding single particle-averages \bar{s}_{sp} and \bar{p}_{sp} (or \bar{r}_{sp}) respectively. In the remainder of this thesis, we will use $r(t)$ to represent *either* anisotropy or polarization as appropriate to the type of analysis used. Also included is the amplitude of the anisotropy or polarization time-amplitude autocorrelation function (TACF) which is defined as:

$$G(0) - G(\infty) = \overline{p^2} - (\bar{p})^2 \quad (2.13)$$

where $G(0)$ is the initial and $G(\infty)$ is the residual or limiting value of the TACF defined as:

$$G_{rr}(\tau) = \lim_{T \rightarrow \infty} \frac{1}{T - \tau} \sum_{t=0}^{T-\tau} [r(t) - \bar{r}_t] [r(t + \tau) - \bar{r}_{t+\tau}] \quad (2.14)$$

This can also be written as:

$$G_{rr}(\tau) = \lim_{T \rightarrow \infty} \frac{1}{T - \tau} \sum_{t=0}^{T-\tau} [r(t)] [r(\infty) + [r(t + \tau) - r(\infty)] e^{-6D_r \tau}] \quad (2.15)$$

Table 2.1: Prediction of polarization TACF amplitudes and other quantities for various combinations of experimental conditions including ensemble and single-particle experiments, one- and two-dimensional absorbers, polarized and non-polarized excitation, and true anisotropy, apparent anisotropy, and polarization. The table has been organized hierarchically by experimental geometry for the first level and ordered by type of excitation for the second in order to facilitate comparison between experiments in which either true or apparent anisotropy or polarization were calculated. The third column lists the type of QD absorption dipole. The single-particle numbers for non-polarized excitation in the case of apparent anisotropy and polarization result from numerical integration, while other calculations involve closed-form integration.

Experimental Conditions			Ensemble measurements			Single-particle measurements			
analysis	exc	abs	15 $\langle I_x, I_y, I_z \rangle$	3 $\langle s \rangle$	$\langle r \rangle$ or $\langle p \rangle$	3 $\langle s_{Sp} \rangle$	$\langle r_{Sp} \rangle$	$\langle r_{Sp}^2 \rangle$	$G(0)-G(\infty)$
true	x	x	3,1,1	1	2/5	1	0	0.267	0.27
true	x	xy(np)	3,1,1	1	2/5	1	0.250	0.231	0.17
true	xy(np)	x	2,2,1	1	0	1	0	0.267	0.27
true	xy(np)	xy(np)	2,2,1	1	0	1	0	0.161	0.16
app	x	x	3,1,1	1	2/5	1	0.121	0.288	0.27
app	x	xy(np)	3,1,1	1	2/5	1	0.300	0.274	0.18
app	xy(np)	x	2,2,1	6/5	0	6/5	0.121	0.288	0.27
app	xy(np)	xy(np)	2,2,1	6/5	0	6/5	0.065	0.150	0.15
pol	x	x	3,1,1	4/5	1/2	4/5	0	0.500	0.50
pol	x	xy(np)	3,1,1	4/5	1/2	4/5	0.293	0.363	0.28
pol	xy(np)	x	2,2,1	4/5	0	4/5	0	0.500	0.50
pol	xy(np)	xy(np)	2,2,1	4/5	0	4/5	0	0.273	0.27

One important observation from Table 2.1 is that non-polarized xy-excitation is predicted to give ensemble and average single-molecule anisotropies of zero. In fact, this is true for both true and apparent anisotropy as well as for polarization and is also true for both one- and two-dimensional absorbers. This can be easily explained by the fact that non-polarized excitation results in equal average fluorescence intensities in the vertical and horizontal orientations (*i.e.* $\langle I_x \rangle = \langle I_y \rangle$) and therefore the average anisotropy is zero ($\langle r_{\text{ens}} \rangle = 0$). The two exceptions to this occur in the case of the single-particle measurement of the apparent anisotropy. This discrepancy occurs because I_y which, in the case of the apparent anisotropy is substituted for I_z , is an imperfect approximation of I_z . In the case of x-polarized excitation, this gives a constant limiting anisotropy of $(3 \times 2^{1/2} - 4)/2 = 0.121$. However, although the average polarization is zero for non-polarized excitation, the polarization (and anisotropy) fluctuation TACF amplitudes ($G(0) - G(\infty)$) are *not* zero and it is from this decay that important information on rotational dynamics can be extracted. Furthermore, comparison of these amplitudes to those corresponding to measurements taken with x-polarized excitation reveals that they are equivalent. Thus it is expected that experiments conducted with non-polarized excitation will not produce significantly different results from those obtained from polarized excitation.

Examination of the Table 2.1 leads to another interesting observation. If the QD is treated as a one-dimensional absorber, the TACF fluctuation amplitude is found to be ~ 0.27 for both true and apparent anisotropy, but, if instead polarization is measured, then this value is increased to ~ 0.50 . A similar trend is observed for two-dimensional absorbers. Thus by measuring polarization there is the added advantage of a larger TACF amplitude and a thus more sensitive measurement. As alluded to previously, in the case of an absence of information on I_z due to the

limitations of the experimental setup, another advantage of measuring polarization is that, unlike apparent anisotropy which gives an average single-particle anisotropy of 0.121, the polarization is, as is the case for the true anisotropy, equal to zero.

There is some debate about whether the QDs that we are using in this study behave as one-dimensional or two-dimensional absorbers. If the aspect ratio is greater than two, then the QD is expected to behave as a one-dimensional absorber with a single absorption transition dipole aligned with the principal C3 axis of the molecule [3]. If the aspect ratio is less than two, then the QD is expected to behave as a two-dimensional absorber. In this case it would be more appropriate to treat them as two-dimensional absorbers with two degenerate orthogonal equatorial transition absorption dipoles which are in turn orthogonal to the C3 principal axis. As previously mentioned, our QD655 were measured by TEM to have dimensions of 5.8 nm x 12.8 nm [5] which would give an aspect ratio of ~2.2. Given this ratio calculated from our TEM measurements is greater than 2, we might expect these QDs to possess a two-dimensional transition absorption dipole, but this hypothesis is tentative because this value is only slightly above two and thus borderline.

Table 2.1 clearly shows that, if the sample is excited by x-polarized light, then the ensemble-averaged intensity functions and polarization (as well as the true and apparent anisotropies) are equivalent regardless of whether the QD is treated as a one-dimensional or a two-dimensional absorber. The average squares of the anisotropies however are consistently lower for two-dimensional absorbers than they are for one-dimensional absorbers. Consequently, because the polarization TACF amplitude is the average square of the polarization subtracted

from the squared average, the smaller average squared polarizations associated with two-dimensional absorbers result in overall reduced polarization TACF amplitudes for these one-dimensional absorbers relative to two-dimensional absorbers. Typically, these values are lowered to $\sim\frac{1}{2}$ - $\frac{2}{3}$ the values observed for one-dimensional absorbers.

Taken together, the above points have important implications for our experiments. The important takeaway for our experiments in which we measure polarization is that, regardless of whether the QD behaves as a one- or two-dimensional absorber, the polarization fluctuation TACF will remain the same for x-polarized and non-polarized excitation. For one-dimensional absorbers, this value is predicted to be about 0.50. For two-dimensional absorbers, it is reduced substantially to ~ 0.27 - 0.28 , but still predicted to be near identical for polarized and non-polarized excitation.

2.2.4. Type I Fc ϵ Receptor

The Type I Fc ϵ Receptor (Fc ϵ RI) is a member of a class of receptors known as Fc receptors and one of two receptor types belonging to the Fc ϵ receptor subclass [16]. Possessing two immunoglobulin-like domains, it also belongs to the immunoglobulin superfamily. It exhibits a high affinity for immunoglobulin E (IgE). The Fc ϵ RI is a modular tetrameric structure (see Figure 2.4) consisting of a single membrane-spanning α -subunit, a quadruple membrane-spanning β -subunit with three loops, two of which are extracellular and one of which is intracellular, and two γ -subunits linked by a pair of disulfide bonds. In rat basophilic leukemia cells as we have used in this study, the α -subunit has a molecular weight of ~ 50 kDa, the β -

subunit ~33 kDa, and the γ -subunits are about 7-9 kDa each, giving a total weight for the receptor of ~97-101 kDa [17]. The α -subunit structure possesses two extracellular domains which together comprise the ligand binding pocket for IgE antibody. It binds to the constant sequences contained with the Fc portion of the ϵ heavy chain of IgE [18]. The α -subunit has two other parts: a transmembrane-spanning chain and a cytoplasmic chain. The β -subunit consists of four transmembrane domains and two extracellular loops. The first extracellular loop is located between the first and second transmembrane domains and the second loop between the third and fourth transmembrane domains. It also contains a cytoplasmic loop between the second and third transmembrane domains along with a carboxy terminus and an amino terminus both on the cytoplasmic side of the membrane. Consistent with its role in initiating downstream signal transduction, the β -subunit contains an immunoreceptor tyrosine-based activation motif (ITAM) located within its carboxy-terminal domain [19]. The disulfide-linked γ -subunits form a homodimer and, as is the case for the β -subunit, play a role in the activation of intracellular messengers (namely tyrosine kinases) which relay information to the interior of the cell through a signaling cascade. Each γ -subunit of the homodimer complex also contains an ITAM located in its cytoplasmic domains [19], consistent with its role in such a process. These ITAMs activate src and syk tyrosine kinases in sequence. The signal transduction pathways initiated by these kinases later connect with pathways that overlap with those of many other receptors.

IgE, one of several immunoglobulin isotypes, is the cognate ligand of the Fc ϵ RI, which it binds with a stoichiometry of 1:1. IgE, like other immunoglobulin antibodies, is comprised of two Fab fragments and two Fc fragments (see Figure 2.5). Two duplicate ϵ heavy chains which are comprised of ~550 amino acids each span both of these regions and are linked by disulfide

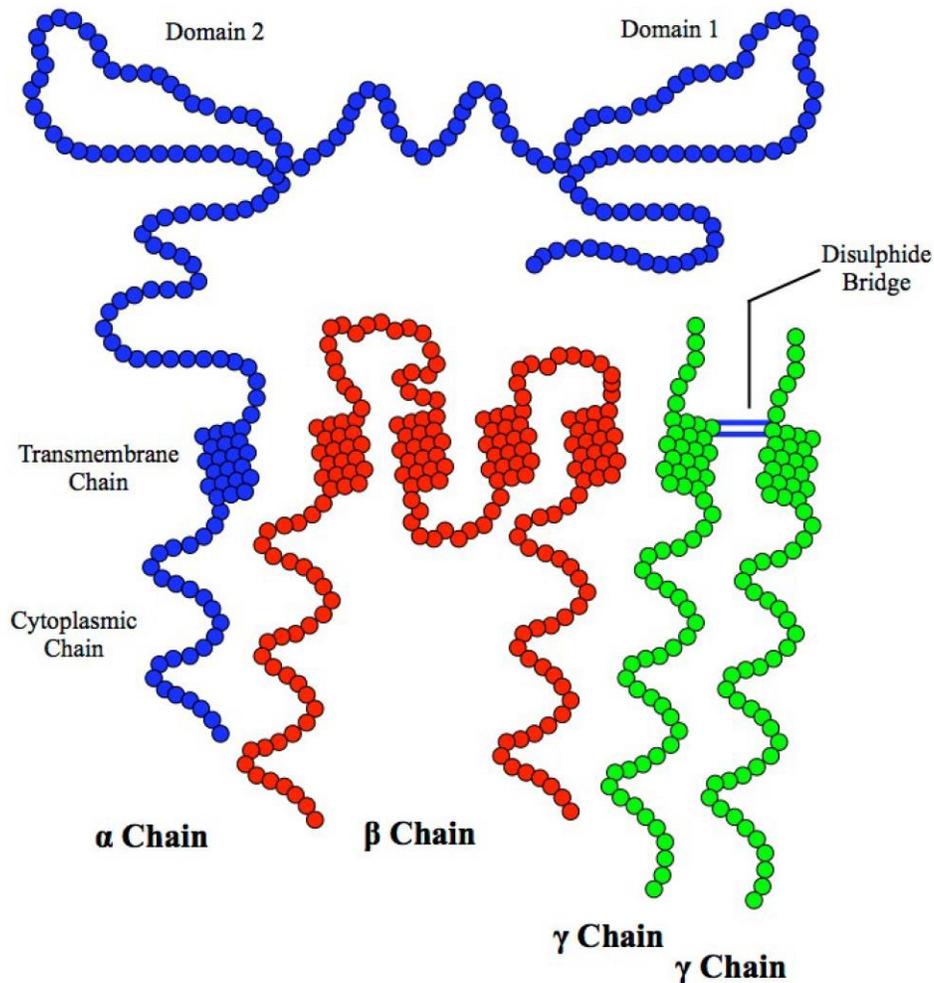


Figure 2.4: Schematic diagram of FcεRI depicting its α , β , and two γ –chains which are bound by intermolecular interactions. The α -subunit structure possesses two extracellular domains which together comprise the ligand binding pocket for IgE antibody. It also possesses a transmembrane-spanning chain and a cytoplasmic chain. The β -subunit consists of four transmembrane domains and three loops, two of which are extracellular and one of which is cytoplasmic. The carboxyl and amino termini of the β -subunit are both located on the cytoplasmic side of the membrane. The γ -subunits each consist of an extracellular domain, a transmembrane domain, and a cytoplasmic domain and are linked by intermolecular disulfide bonds. Image adapted from Sari, 2011 [20].

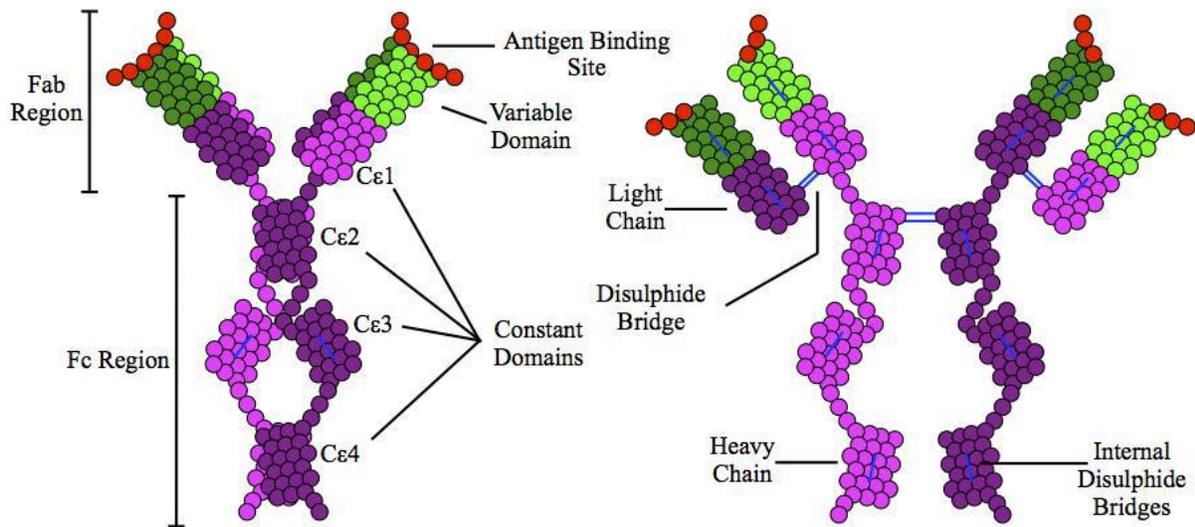


Figure 2.5: Schematic diagram of IgE antibody depicting its two heavy chains and two light chains viewed from the side (left) and rotated 90 degrees (right). The heavy chains span both the Fc and Fab region whereas the light chain spans only the Fab region. The Fc portion of each heavy chain consists of three constant domains and the Fab portion possesses a single constant domain and terminates in a variable domain. Each light chain possesses a constant domain and terminates in a variable domain. Together the four variable domains comprise the antigen binding site of the antibody. Image adapted from Sari, 2011 [20].

bonds [21]. The ϵ heavy chain defines the IgE immunoglobulin subclass. Two equivalent light chains spanning the Fab region are joined to the heavy chains by disulfide bonds. Each of the heavy chains has a molecular weight of ~50 kDa and the light chains have molecular weights of ~25 kDa giving a total molecular weight of approximately 150 kDa for the entire IgE antibody [21]. The Fc portion of the heavy chain is comprised of three constant segments (C_{H4} , C_{H3} , and C_{H2}). The Fab portion of the heavy chain consists of a constant segment (C_{H1}) and a variable segment (V_H). Each light chain consists of a constant segment (C_L) and terminates in a variable segment (V_L). The V_H and V_L segments together comprise the antigen binding site of the antibody. As the variable designation implies, these segments confer antigen binding specificity to the antibody by recognizing only the epitope of a particular antigen.

The Fc ϵ RI plays an important role in the adaptive immune system and is involved in allergic responses to pollen and other allergens. This receptor is expressed on epidermal Langerhans cells, eosinophils, mast cells, basophils, and antigen-presenting cells [22], [23]. It is constitutively expressed only on mast cells and basophils [24]. Upon crosslinking of Fc ϵ RI-bound IgE expressed on mast cells by polyvalent antigen, a signal transduction cascade is triggered leading to mast cell degranulation. The degranulation process involves the exocytosis of vesicle contents releasing a variety of compounds into the extracellular milieu. These compounds, which include the small vasoactive amines histamine and serotonin, proteins such as proteoglycans and serine protease enzymes, and cytokines, interleukins, leukotrienes, and prostaglandins amongst other agents, mediate the inflammatory response characteristic of allergy [25, 26]. This produces a variety of physiological changes including mucous secretion, vasodilation, and smooth muscle contraction which are responsible for symptoms associated with

allergy such as rhinorrhea, pruritis, dyspnea, and anaphylaxis. These symptoms may manifest in an entire-body, systemic response such as observed in anaphylaxis, or a localized response that affects a single body system such as asthma which is localized to the respiratory system and involves inflammation of the bronchi and bronchioles [21].

This process of FcεRI activation typically occurs within the context of an acute allergic response known as a type I hypersensitivity reaction in which mast cells or basophils are primed for an allergic response after initial exposure to an antigen/allergen. These steps precede the receptor crosslinking and subsequent cell degranulation steps described above. Upon initial exposure to an allergen, the allergen is bound and displayed by antigen-presenting cells. The allergen is then presented to a type of T-helper cell known as a T_H2-lymphocyte. Through direct interaction and production of the cytokine interleukin-4, these T_H2 cells stimulate another type of lymphocyte known as bursa- or bone marrow-derived cells (B cells) to produce IgE antibody which is released into systemic circulation. Upon encounter with a mast cell or basophil expressing FcεRI on its surface, the IgE binds the receptor and the cell is now considered to be sensitized to the allergen. Re-exposure to allergen initiates the aforementioned receptor crosslinking and subsequent cell degranulation processes which lead to the variety of physiological changes associated with allergy some of which are described above [21].

2.2.5. Goals of Study

The major goal of the study contained herein was, firstly, to demonstrate the feasibility of measuring cell surface receptor rotation through optical examination of QD-labeled receptors.

We then applied this method to examine the effect of a variety of cell treatments expected to perturb receptor rotation either via changing receptor interactions or altering the local environment of the receptor. Including a control group, we conducted experiments with six such treatments. With the exception of the control group which was only tested with non-polarized excitation, all groups had a further three subgroups which consisted of two groups measured with non-polarized excitation and one with polarized excitation. We structured our experiments in this manner in order to facilitate assessments of the reproducibility of our measurements and to test our hypothesis concerning the equivalence of polarized and non-polarized excitation. The control group, hereafter referred to as the “dry” group, consisted of untreated QDs fixed on glass coverslips and these QDs were as such expected to remain immobile. The group treated with IgE only was the cell-bound control group. In the absence of foreign antigen to crosslink IgE-bound receptor, we expect to see rotational motion corresponding to solely the IgE-bound Fc ϵ RI unit/complex. Our group has previously monitored the dynamics of erythrosin-tagged IgE-Fc ϵ RI expressed on 2H3 cells using time-resolved phosphorescence anisotropy and it was indeed observed to exhibit a rotational correlation time of 82 μ s which was taken to reflect the hydrodynamic rotation of the receptor-ligand complex [27]. The other four treatments involved IgE and an additional agent. Receptors expressed on cells treated with dinitrophenyl bovine serum albumin (DNP-BSA), an agent consisting multiple dinitrophenyl haptens linked to a protein carrier, were expected to exhibit slowed rotation relative to the IgE-only group because this agent crosslinks receptor-bound DNP-specific A2 IgE antibody [27]. Given that paraformaldehyde crosslinks membrane proteins, cells treated with this agent were expected to undergo an even more drastic reduction in rotational motion to the point of immobility [28]. The potential effects of M β CD and cytochalasin D, agents which deplete membrane cholesterol and

disrupt cytoskeletal integrity respectively, on FcεRI rotation are more complex and especially interesting. These selected treatments provide feedback on the viability of our method and the results of these experiments have important implications for FcεRI receptor function.

2.3. Materials and Methods

2.3.1. Wet Chemistry and Cell Culture Procedures

Quantum Dots 605 and QD655 were obtained from Invitrogen, a subsidiary of Life Technologies (Eugene, OR). QDs were conjugated to streptavidin, a protein with a high affinity for biotin. To prepare the non-cell-bound dry QDs on glass which were used as a control group, after ligation of streptavidin to the QD, one μmol of these streptavidin-bound QD655 was dissolved in a liter of ethanol and diluted in ethanol to a final concentration of 0.005 nM. A 20 μL aliquot was pipetted on a glass coverslip with a spin coater operating at 3000 rpm. The glass coverslips were then mounted on the microscope slides.

2H3-RBL cells were cryogenically preserved in liquid N₂ with DMSO as a cryoprotectant. Cells were partially thawed in a water bath and transferred into Minimum Essential Medium. Cells were centrifuged with a vortexer for 4 min. and the supernatant removed to extract any residual DMSO contaminant. The cells remained in the pellet and this cycle was repeated two additional times. Media was again added to the cells along with 10% fetal bovine serum and 200 nM L-glutamine as nutritional supplements and 1x10⁴ units/mL penicillin G, 10 μg/mL streptomycin, and 25 μg/mL fungizone as antibiotics/antifungals to inhibit contamination by

other microorganisms. The cells were kept in culture flasks and 1-1.5 days before experiments were detached from the sides of the flask using the chelating agent EDTA, transferred to petri dishes using sterile technique, and allowed to attach to the bottom of the dish and grown to ~50% confluence as described previously [29]. A Bunsen burner was used to sterilize all potentially contaminating surfaces. Cells were maintained in an incubator under humidified conditions with a concentration of 5% CO₂ and at a temperature of 37° C.

Cells were *sensitized* with biotinylated DNP-specific IgE antibody designated A2 which was kindly provided by Dr. Israel Pecht of the Weizmann Institute of Science, Rehovoth, Israel and then subsequently treated with quantum dots conjugated to streptavidin, a protein with high affinity for biotin. The streptavidin functioned as the linker between the quantum dot and the receptor-bound biotinylated IgE. Cells were washed twice with a 1 mL pH 7.4 PBS/0.1% BSA buffer solution and subsequently incubated with 1 nM biotinylated IgE for 20 minutes. After cells were labeled, they were washed four times with the aforementioned buffer solution. Accompanying directions from Invitrogen regarding conjugation of these quantum dots to biotinylated proteins were followed in order to conjugate QD to protein in a stoichiometric ratio of one or less [30].

A portion of *samples were treated* with 100 µg/mL DNP-BSA in PBS for 1 h, another portion with 40 mg/mL paraformaldehyde in PBS for 45 min., another with 100 µg/mL MβCD in PBS for 1 h, and another with 40 µM (~20.3 µg/L) cytochalasin D in PBS for 1 h as performed previously [5]. After their respective treatments, cells were washed four times with the previously mentioned buffer solution and incubated with 100 pM streptavidin-conjugated QD655

for 10 min. Before an experiment, all medium was thoroughly removed, washed several times with buffer solution to remove any residual fluorescent phenol red, and replaced with buffer solution.

2.3.2. Imaging Collection and Analysis

Images were collected with a Zeiss Axiovert 200M microscope with a Zeiss α Plan-Fluar oil immersion objective at 100x magnification and a numerical aperture of 1.40. The microscope 2.5x Optivar and a 2x image extender/magnifier yielded a total magnification of 500x at the camera chip. The microscope was outfitted with a 460 +/-50 nm bandwidth excitation filter, a 475 nm cutoff dichroic mirror and a 655 +/-40 nm bandwidth emission filter. A 100 W arc lamp was used to excite the sample. A schematic of the apparatus is shown in Figure 2.6. For one out of the three groups of experiments conducted for each treatment, vertically-polarized light was used to excite the sample instead of non-polarized light. A Princeton Instruments Dual View image divider equipped with a polarizing beamsplitter enabled the simultaneous collection of vertically- and horizontally-polarized images for each frame. An Ixon Andor DU897E EMCCD camera with 512x512 pixel resolution and a 16x16 μm pixel size was used to collect images. Images were collected at a frame rate of 10 ms/frame with 1000 frames recorded in total. To achieve that speed, the 512x512 chip was binned to 102x102 so our actual images are 102x102. A sample image of cell-bound QD is shown in Figure 2.7. Fluorescence is observed on the periphery of the cell corresponding to QD-labeled receptor localized to the cell membrane whereas the interior of the cell does not fluoresce in this manner except for some autofluorescence emanating primarily from the nucleus. The dry quantum dots mounted on glass

coverslips were imaged in the same manner (see Figure 2.8). Background images necessary for later correction of vertical and horizontal intensities were collected by taking images of regions on the cell in which no quantum dot-bound receptor was present.

Vertically and horizontally-polarized image stacks were analyzed using NIH ImageJ. Images were re-expanded 5x to 510x510 pixels, each pixel width being 0.16 μm . In order to perform correlation calculations on separate image frames, it was necessary for the images to be aligned with sub-pixel precision. Since this cannot be accomplished by mechanical optimization of the image alignment via the Dual View image splitter, further correction for this displacement using computational methods was necessary. Image stack pairs were corrected in ImageJ for relative displacement, rotation, and dilation using four adjustable parameters and maximizing the intensity cross-correlation of the two images.

Usually, >10 QDs were selected per cell. In order to compensate for background noise originating from the sample or artifactually-introduced by our camera, it was necessary to conduct various background measurements. The images for these background regions of interest were processed in a similar manner as described for the QD measurements above. A total of four backgrounds were taken per cell and these four vertical and horizontal intensities were separately averaged. The cell backgrounds were all taken in inconspicuous portions of the cell surface where no QD-labeled receptor was present and varied from cell to cell. This enabled correction for factors such as cellular autofluorescence originating from inside the cell. Also calculated was a camera background, which, in contrast to the orientation-dependent backgrounds, does not

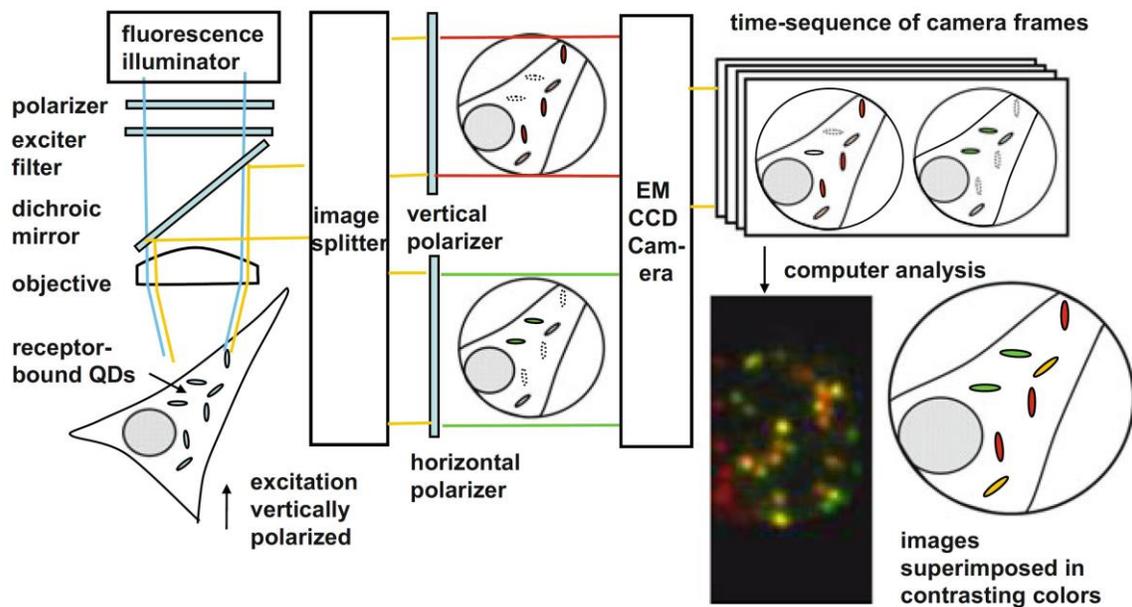


Figure 2.6: Schematic diagram of the imaging apparatus used for measurement of QD rotation on ms timescales. A beamsplitter was used to separate vertically- and horizontally-polarized fluorescence emanating from differently-oriented QDs. Vertically- and horizontally-polarized QD are shown in green and red respectively. Image adapted from Zhang *et al.*, 2017 [5].

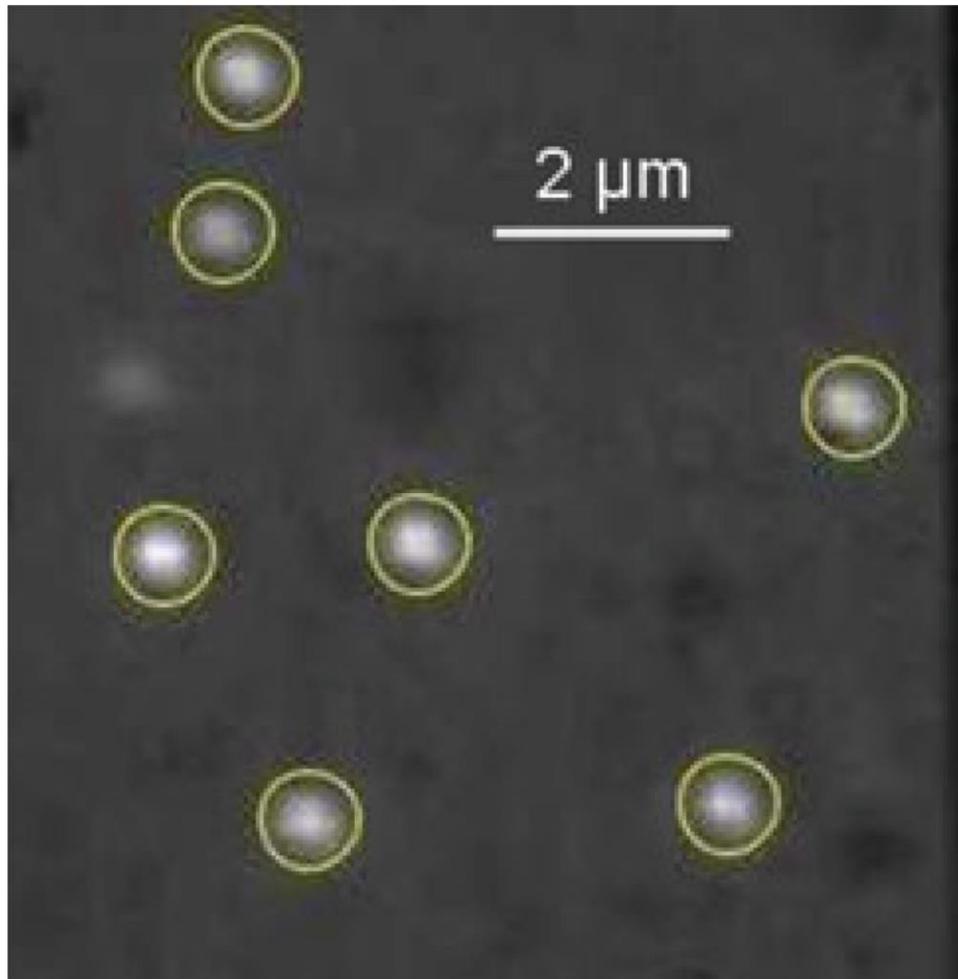


Figure 2.7: Fluorescence image of QD655-IgE bound to FcεRI on 2H3-RBL cells. Image adapted from Zhang *et al.*, 2017 [5].

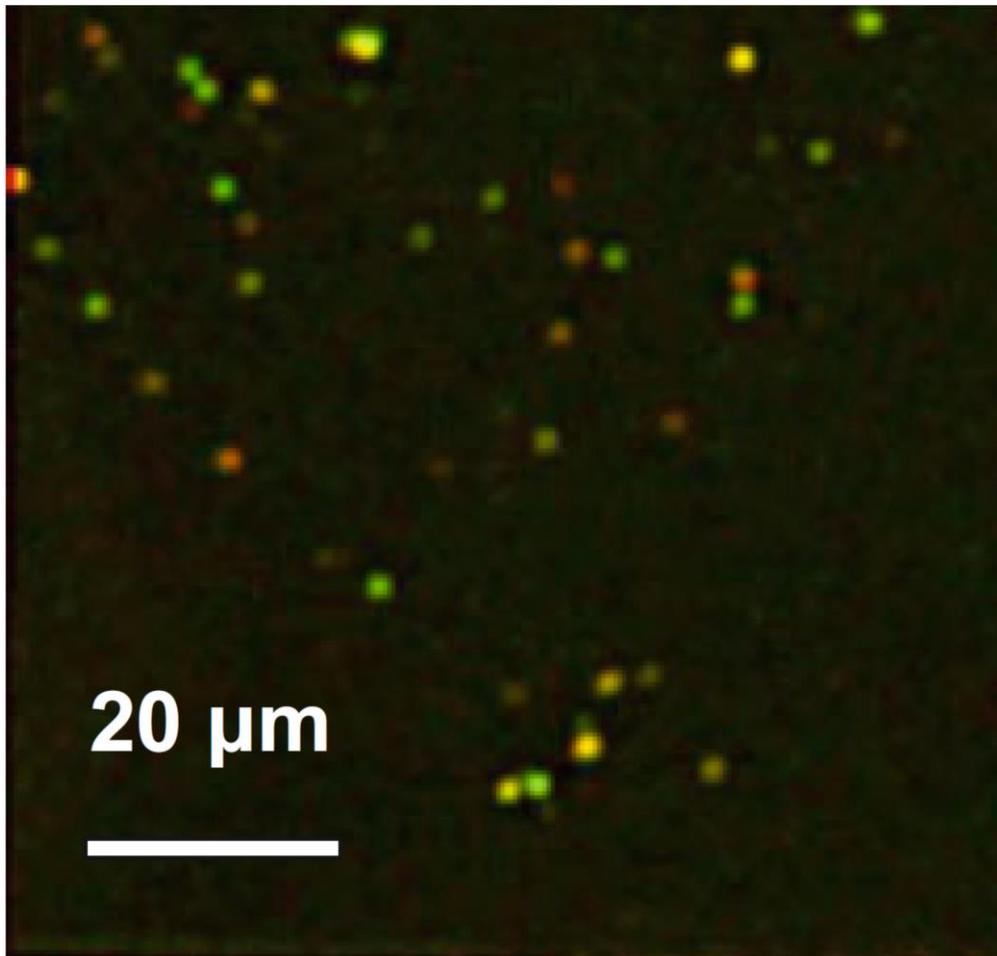


Figure 2.8: Fluorescence image of dry QD605. QD producing vertically- and horizontally-polarized fluorescence are shown in green and red respectively. Image adapted from Zhang *et al.*, 2017 [5].

arise from the sample but instead reflects a constant of ~300 intensity units per frame added by the camera driver. All of these background corrections were necessary because they were of significant magnitude relative to the fluorescence signal emanating from the quantum dots, and failure to account for these would result in the signal being drowned out entirely by background noise.

The final output from this initial processing of the image data consisted of CSV files designated “LRB” to indicate “left, right & background”. Each of these files contained information corresponding to all dots on a single cell. These files consisted of tabulated data of left- and right-hand intensities plotted against the number of frames recorded (typically 1000) for each dot analyzed on a given cell. Also calculated in these spreadsheets were averages for all vertically- and horizontally-polarized intensities.

The LRB files generated by our initial processing of the imaging data were then analyzed in a Mathematica notebook, *i.e.* a .nb program. A major interest was to compare the results from our most recent version of the primary analysis program version 140 (which is included in its entirety as Program Code 3: “Image Corr v.140” in the Appendix) to that of an earlier primary analysis program (see Program Code 1: “Image Corr Mod15k”). We also analyzed results with another intermediary version of the program produced in the process of developing the most recent program which had several notable and significant differences from this final program we settled on (see Program Code 2: “Image Corr v.122”). Though these programs differed slightly in the details of the analysis, they shared the same overall goals and general format. The differences

between this final program from the initial in terms of analysis methods and the results obtained will be discussed subsequently. What follows below is a general discussion of the layout and purpose of this program with a focus on our latest analysis.

Regardless of the version used, the ultimate goals of these programs were the same, namely to go from the uncorrected vertical and horizontal intensities obtained from our initial processing of the imaging data and, from here, to obtain fitted intensity and polarization time-autocorrelation fluctuation functions (TACFs) along with tabulated fitted parameters for these decays such as the intensity and polarization, TACF amplitudes, and correlation times, amongst others. These programs were similar in form and consisted of ten separate sections. The first two were “Definitions” and “Depend” sections which are run once together to initialize the program prior to running subsequent sections of the program. The third section was the “Optimize” section which used a non-linear least squares regression to optimize a number of adjustable parameters used to correct the observed vertical and horizontal intensities for a number of factors which need to be accounted for such as cell and camera backgrounds and differential detection efficiencies in the two orientations. This was done in a manner that allowed us to minimize the dependence of the anisotropy on the intensity function and then, using these parameters, we calculated the intensity and anisotropies point-by-point. From these, the intensity and polarization TACFs were in turn calculated. The next section was “ShowOptResults” which displayed graphs of the TACFs obtained from the aforementioned “Optimize” section. The next two sections were the “FitAnisDecay” and “FitIntenDecay” which fit the aforementioned intensity and anisotropy TACFs and output the resulting graphs and fitted parameters. These graphs and numbers were saved to files using the “SaveResults” and “SavePlots” sections

respectively. The ninth section was the “ProcessOneDot” section, which, after initialization via sections one and two above, ran sections three through eight together in concert and produces the aforementioned results and plots files for a single dot. The tenth and final section of the program was the “DataImport” section, which performed an automated analysis of all the dots for all cells. This section consisted of two sub-sections: the first was entitled “ProcessDotsThisFile” which runs sections 3-8 of the program for all dots for a given LRB CSV file corresponding to a single cell. The second sub-section, entitled “ProcessAllDotsAllFiles” iterated this procedure for all LRB files/cells in a given directory and the “DataExport” function outputs results and plots files as described above for each dot on each cell. The sections that follow elaborate on each of the previously described sections in the Mathematica program in more detail.

The key functions in the program associated with the execution of the tasks described in this section are primarily contained within the “Optimize” and “ShowOptResults” with some code contained in the “Definitions” and “Depend” sections described above. Observed intensities in the vertical and horizontal orientations obtained experimentally had to be corrected using three adjustable parameters. One of these parameters was the g-factor which was used to calibrate intensity measurements so as to account for differential detection efficiencies in the two orientations. It is typically expressed as an adjustment factor for the perpendicular (or in our case horizontal) intensity. With this correction, the polarization is rewritten as:

$$p(t) = \frac{I_v(t) - g_{\text{trrad}} I_h(t)}{I_v(t) + g_{\text{trrad}} I_h(t)} \quad (2.16)$$

where g_{trad} is the traditional, conventionally-used g-factor. The other two parameters were the sample background b and the camera background c . The conversion was then done using the following formulae for the corrected true intensities v_t and h_t in terms of the observed intensities v_{obs} and h_{obs} :

$$v_t = v_{\text{obs}} - c - b/2 \quad (2.17)$$

$$h_t = g_{\text{trad}}(h_{\text{obs}} - c) - b/2 \quad (2.18)$$

For computational reasons we have found it more convenient to use a g-factor defined in the following way:

$$g = \frac{g_{\text{trad}} - 1}{g_{\text{trad}} + 1} \quad (2.19)$$

Re-arranging this equation for g_{trad} , substituting in the equation for h_t above, and multiplying both this term and its respective counterpart in v_t by $(1-g)$ gives the following symmetrical expressions:

$$v_t = (1-g)(v_{\text{obs}} - c) - b/2 \quad (2.20)$$

$$h_t = (1+g)(h_{\text{obs}} - c) - b/2 \quad (2.21)$$

As can be observed, the formula for the corrected intensities involves both a variable term $(1-g)v_{\text{obs}}$ or $(1+g)h_{\text{obs}}$ which varies with the respective observed intensity and a constant expression $(1\pm g)(\pm b/2)-c$ containing contributions from both the cell and camera backgrounds. Careful selection of these parameters was important, because otherwise there could be feed-through of the intensity TACF into the polarization TACF. Since the intensity TACF intrinsically fluctuated in accordance with QD blinking, if the anisotropy TACF was dependent on the intensity TACF, then it too could be adversely affected by such a complication which would compromise the quality of the rotational parameters we hoped to extract from it. To select these parameters in a manner that insured maximum independence between the two quantities, we employed the Marquardt iterative non-linear optimization algorithm. What follows is a derivation of b , g , and the associated variances of v_t and h_t in terms of the observed intensities and their associated variances, quantities that are determined experimentally.

Given that polarization or anisotropy values are small ($\ll 1$), we have for their averages:

$$\overline{v_t} \sim \overline{h_t} \quad (2.22)$$

Equating the averages of v_t and h_t using Equations 2.20 and 2.21 and solving for b gives:

$$b = (1-g)\overline{v_{\text{obs}}} - (1+g)\overline{h_{\text{obs}}} \quad (2.23)$$

And the expression for the corresponding error in b can be determined by the appropriate error propagation formula and set to zero:

$$(1 - g)^2 s_{v_{obs}}^2 - (1 + g)^2 s_{h_{obs}}^2 = 0 \quad (2.24)$$

Rearranging we obtain:

$$g_{trad} = \frac{s_{v_{obs}}}{s_{h_{obs}}} \quad (2.25)$$

The variance of v_t is equal to the average of the squares of v_t minus the square of the average:

$$s_{v_t}^2 = \sum v_t^2 / n - (\sum v_t / n)^2 \quad (2.26)$$

Substituting the expression for v_t from Equation 2.22 into Equation 2.29 gives the following equation:

$$s_{v_t}^2 = (1 - g)^2 s_{v_{obs}}^2 \quad (2.27)$$

Analogously, using the expression for h_t from Equation 2.23 we obtain:

$$s_{h_t}^2 = (1 + g)^2 s_{h_{obs}}^2 \quad (2.28)$$

As a method for assessing whether our method for selecting these parameters was appropriate or not, we quantified the degree to which it achieved our goal of minimizing the

interdependence of the intensity and polarization TACFs. To accomplish this, we first made a provisional assumption that the two quantities were uncorrelated and attempted to determine the chance that we would obtain a non-zero correlation coefficient (*i.e.* determine that the two quantities are in fact correlated) given this assumption. In practice, we calculated “minus log total probability” which was defined as the negative logarithm of the probability that the two quantities were uncorrelated. This was calculated as a function of “minus log probability”, the negative logarithm of the probability of getting a negative vertical or horizontal intensity. We sought to minimize this quantity so that we could accept the null hypothesis and confirm that the intensity and polarization were indeed uncorrelated.

This portion of the analysis procedure was also one of the areas in which there were substantive differences between versions 122 and 140 of the new analysis program. In the “Image Corr v.122” of the program, g, b, and c were all optimized together whereas in “Image Corr v.140”, g and b were determined in closed form from background measurements and then the third parameter c was adjusted in the fit.

The fluctuation time-autocorrelation functions of the intensity and polarization fluctuations were calculated using the following formulae applicable to small fluctuations in s or p, respectively:

$$G_s(\tau) = \frac{1}{T} \sum_{t=0}^{T-\tau-1} [s(t) - \langle s_t \rangle][s(t+\tau) - \langle s_t \rangle] \quad (2.29)$$

$$G_p(\tau) = \frac{\frac{1}{T} \sum_{t=0}^{T-\tau-1} [p(t) - \langle p_i \rangle][p(t+\tau) - \langle p_i \rangle] w(t) w(t+\tau)}{\sum_{t=0}^{T-\tau-1} w(t) w(t+\tau)} \quad (2.30)$$

where τ is the correlation time in frames, T is the total number of frames, and $G_s(\tau)$ and $G_p(\tau)$ are the intensity and polarizations TACFs respectively. It is important to note that these averages are over regions consisting of the same number of points

The intensity and polarization fluctuation TACF decays were both fitted with a two-parameter single-exponential where the two parameters were an amplitude $G(0)$ and a correlation time τ using a Marquardt non-linear least-squares regression. In some experiments, a three-parameter fit which included an additional constant, a limiting polarization correlation $G(\infty)$ was used to fit the polarization fluctuation TACF. Given that the polarization fluctuation TACF should decay to zero, $G(\infty)$ can then be considered a measure of imperfect fit. The resulting fitted parameters were saved in results CSV files with one file corresponding to each individual dot. The various graphs obtained, including plots of the intensity and anisotropy functions and TACFs as well as plots fitted for 100, 200, 500, and 1000 (all) points, amongst others, were saved in PDF files.

Results CSV files output from the final primary analysis program containing data for each quantum dot were processed using a program written in Visual Basic to merge data. This program was used to merge intensity and polarization TACF functions for all dots in a given group into a single file for comparison of individual dot results.

Fitted parameters from the results files generated by the final primary analysis program corresponding to each quantum dot were combined and averaged by a secondary notebook analysis program in Mathematica (see Program Code 4: “MergeFiles v.39” in Appendix). Weighted arithmetic averages of the polarization TACF amplitudes $\langle G_p(0) \rangle$ were calculated as follows:

$$\langle G_p(0) \rangle = \frac{\sum_{i=1}^n w_i [G_p(0)]_i}{\sum_{i=1}^n w_i} \quad (2.31)$$

where n is the total number of dots and $[G_p(0)]_i$ is the polarization fluctuation TACF amplitude of the individual dots i . The weights were calculated as:

$$w_i = \frac{1}{\sigma_{[G_p(0)]_i}^2} \quad (2.32)$$

the accompanying weighted linear standard deviation was calculated using the following formula:

$$\sigma_{G_p(0)} = \sqrt{\frac{N \sum_{i=1}^n w_i ([G_p(0)]_i - \langle G_p(0) \rangle)^2}{(N-1) \sum_{i=1}^n w_i}} \quad (2.33)$$

Because the range of rotational correlation times spanned several orders of magnitude, a weighted geometric average was calculated instead of an arithmetic average as follows:

$$\bar{\tau}_{wtd} = \exp \left(\frac{\sum_{i=1}^n w_i \ln \tau_i}{\sum_{i=1}^n w_i} \right) \quad (2.34)$$

where, if the value and standard deviation of a rotational correlation time for a particular quantum dot are both greater than zero, then the weight corresponding to this is inversely proportional to the variance of the natural logarithm of the rotational correlation time:

$$w_i = \frac{\tau_i^2}{\sigma_{\tau_i}^2} \quad (2.35)$$

The accompanying weighted geometric standard deviation was determined using the following formula:

$$\sigma_{\tau} = \bar{\tau}_{wtd} (e^{\sigma_{\ln \tau}} - 1) \quad (2.36)$$

where the standard deviation of the natural log of the rotational correlation time was calculated as:

$$\sigma_{\ln \tau} = \sqrt{\frac{N \sum_{i=1}^n w_i (\ln \tau_i - \langle \ln \tau \rangle)^2}{(N-1) \sum_{i=1}^n w_i}} \quad (2.37)$$

Tabulated parameters for each quantum dot along with averages for these parameters output from the secondary Mathematica analysis program and graphs of TACF intensity and polarization decays output from the Visual Basic program were compiled together in Excel spreadsheets. Quantum dots were screened for quality using a number of constraints. Typically, dots were excluded if they gave parameters that did not fall within the following constraints: a range of 0.2-0.6 for the g-factor, a range of 0-0.01 for the amplitude of the polarization fluctuation TACF, a range of 1-200 μs for the rotational correlation time, and 0-0.0012 for the standard deviation of the anisotropy TACF. Additionally, QDs which passed this initial screening were visually inspected and manually excluded if they exhibited abnormal behavior in either the intensity or polarization TACF. A secondary Mathematica analysis program used to merge and average the results files for each individual QD was then used to obtain averages and associated statistics for these remaining select QDs.

2.4. Results and Discussion

2.4.1. Exponential Decay of Polarization Fluctuation TACF is due to Rotational Reorientation and not Blinking

A major concern pertaining to these measurements was whether QD blinking would be a confounding factor that contributed to the decay of the polarization fluctuation TACF. If this were the case, this would represent a significant issue because then the decay would not be solely attributable to rotational reorientation because of the feed-through of blinking into the polarization fluctuation TACF. A comparison of cell-bound QD655 to dry QD655 largely allays these concerns (see Figures 2.9 and 2.10). Both cell-bound and dry dots blink yet, while dry dots are largely immobile and display correspondingly flat decay curves (except for shot noise in the first channels giving a variable non-zero $G_p(0)$), the cell-bound dots exhibit the exponential decay characteristic of rotational reorientation. The presence of this difference in the polarization fluctuation TACFs of cell-bound and dry dots, despite these differently-treated groups being similarly affected by blinking, suggests that the fluctuations in intensity due to blinking did not affect the polarization fluctuation TACF to a significant extent. A further argument against such a complication comes when the intensity fluctuation TACF and the polarization fluctuation TACF for the cell-bound QD are examined in parallel (see Figures 2.11 and 2.12). This indicates that, while the kinetic signature characteristic of blinking phenomena is present in the intensity fluctuation TACF, this is not the case for the polarization fluctuation TACF.

2.4.2. Lateral diffusion does not distort polarization fluctuation TACF

Lateral diffusion of receptors represents another possible complication that could potentially interfere with our measurements. This could have an effect if the regions of interest selected around the QD for the vertical and horizontal images were displaced relative to each other. If such misalignment were present, lateral diffusion resulting in the movement of the QD

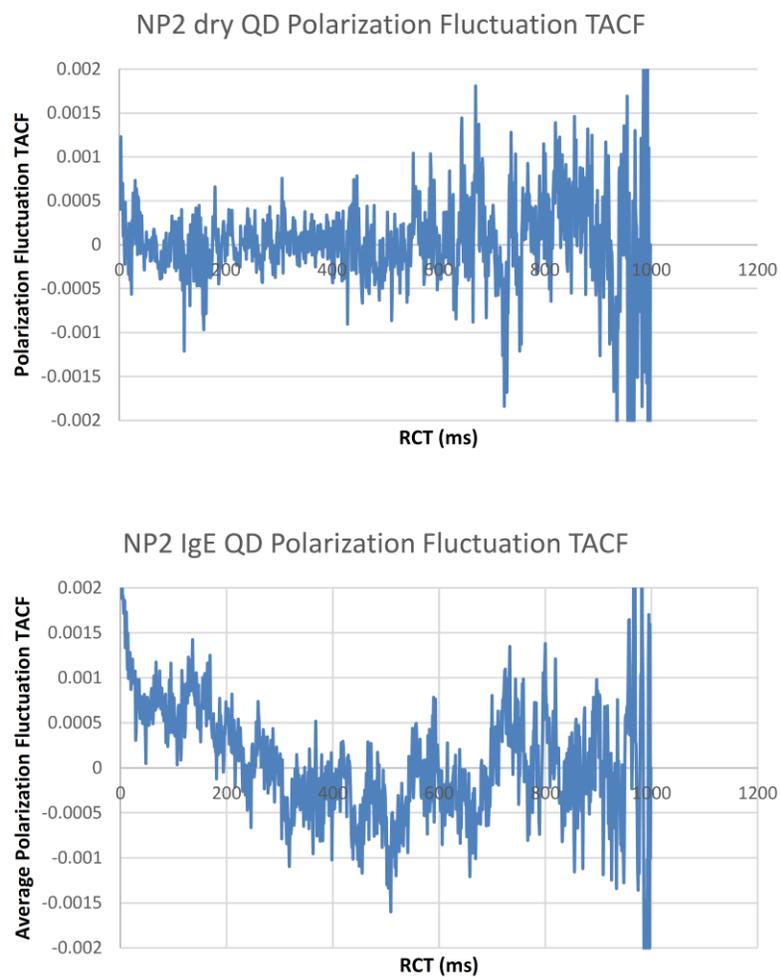


Figure 2.9: Polarization fluctuation TACF for a single QD from NP2 dry subgroup (top panel) and from the NP2 IgE subgroup (bottom panel). The gradual decay of the IgE only-treated QD compared to the immediate decay of the immobile dry QD on glass suggests that the decay is due to actual rotational reorientation and not an artifact of QD blinking.

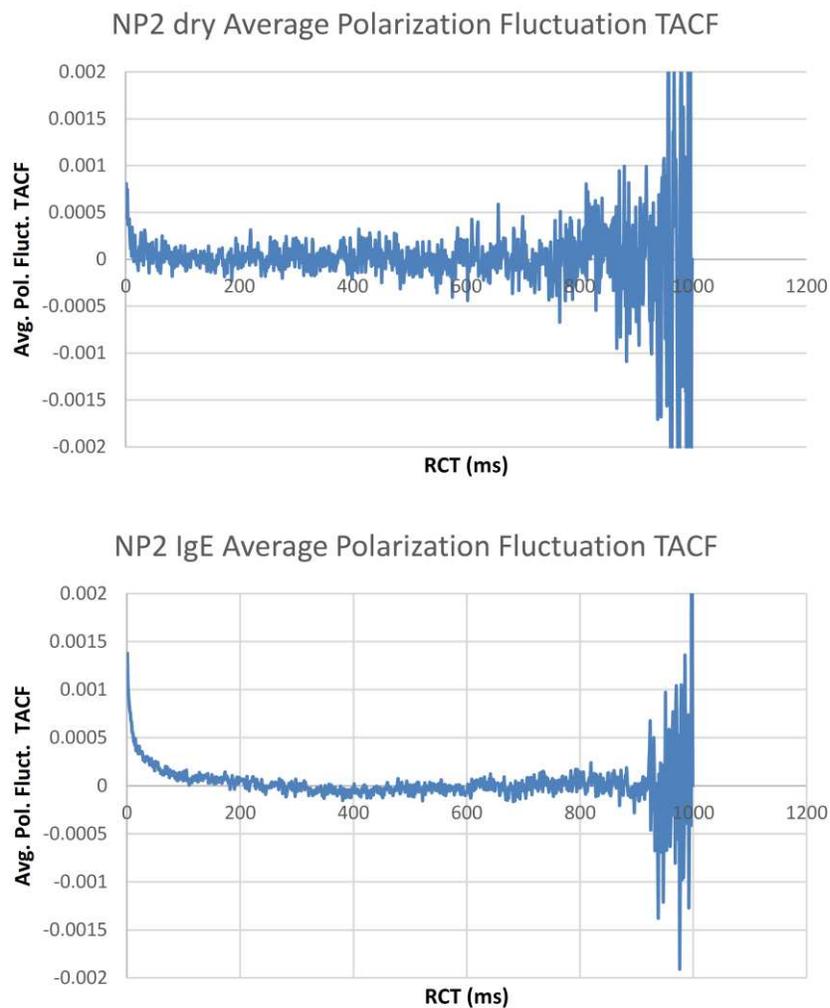


Figure 2.10: Average polarization TACF for NP2 dry subgroup (top panel) and average polarization fluctuation TACF for NP2 IgE subgroup (bottom panel). The gradual decay of the NP2 IgE QDs as compared to the immediate decay of the immobile NP2 dry dots after the first channel suggests that the NP2 IgE decay is due to rotational reorientation and not QD blinking.

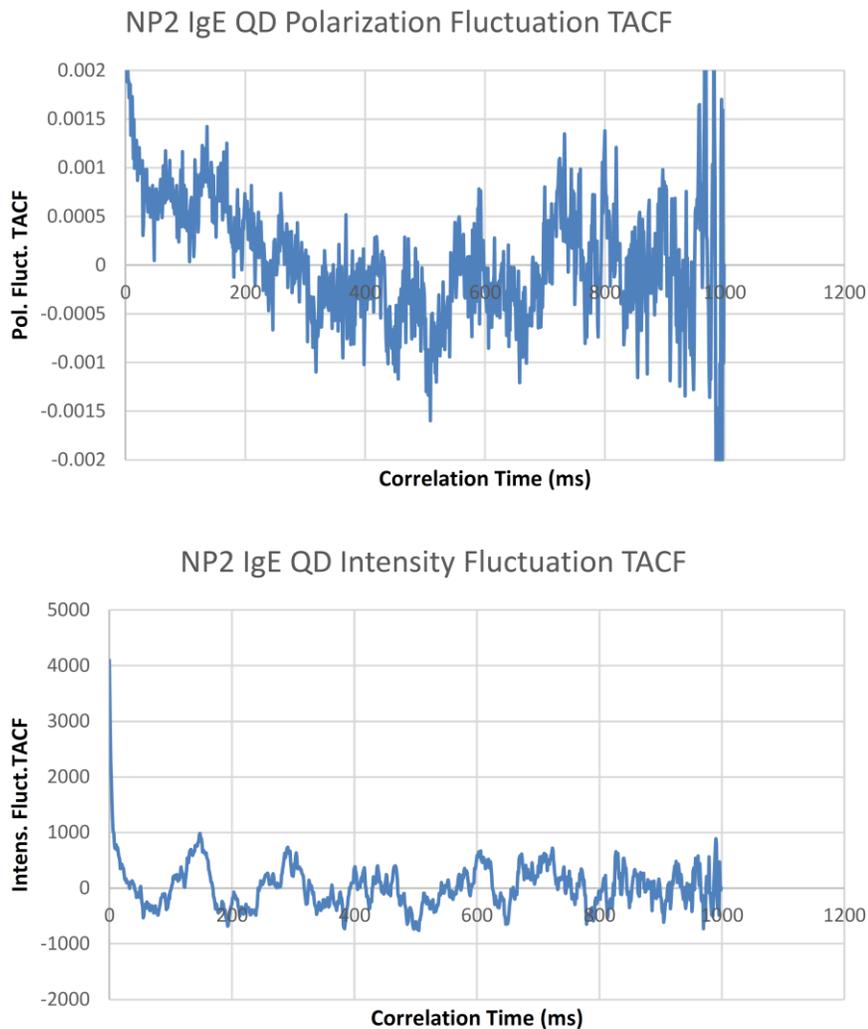


Figure 2.11: Polarization fluctuation TACF (top panel) vs intensity fluctuation TACF (bottom panel) for a single QD in the NP2 IgE subgroup. The large fluctuations between non-zero and zero intensity in the intensity fluctuation TACF are indicative of a QD switching between an “on” state to an “off” state. This kinetic signature is not readily apparent in the polarization fluctuation TACF.

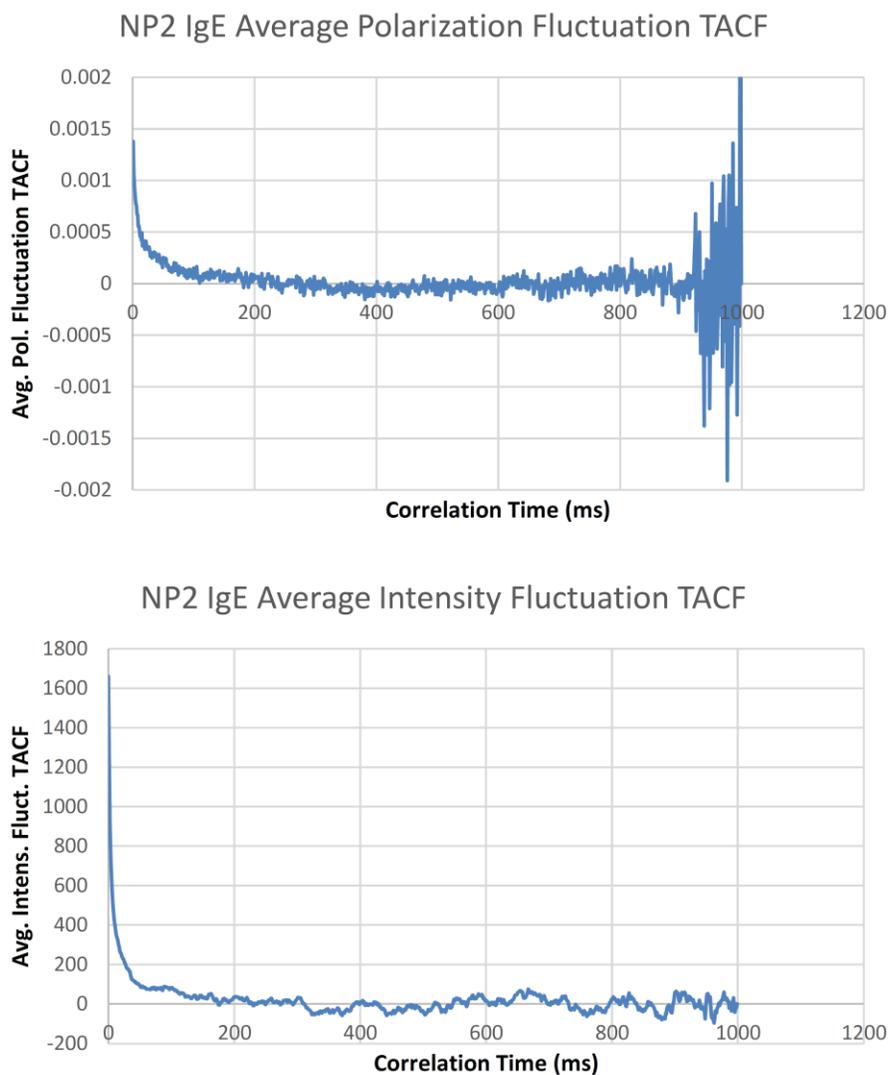


Figure 2.12: Average intensity fluctuation TACF (top panel) and average polarization fluctuation TACF for NP2 IgE subgroup (bottom panel). Whereas the kinetic signature of QD blinking is readily apparent in the intensity fluctuation TACF, no such phenomenon is reflected by the polarization fluctuation TACF.

from an image corresponding to one polarization to the image corresponding to the other polarization would artificially decrease the signal in the former channel and inflate the signal in the latter, distorting the apparent polarization fluctuation TACF.

Fortunately, experimental evidence suggests that lateral diffusion does not affect our measurements to an appreciable extent. One piece of evidence comes from comparisons of data sets taken for QDs with differently-sized regions of interest (ROIs). For one set of data, a 2x extender was used in addition to a 2.5x Optivar which, with the 100x objective, gave a total magnification of 500x. The other set of data was collected without the extender, giving a total magnification of 250x.

Additionally, paraformaldehyde fixation did not produce a significant change in the TACF decay relative to cells treated solely with IgE (see Figures 2.13 and 2.14). Paraformaldehyde is a protein cross-linking agent and is thus expected to result in complete lateral immobilization of the receptor [28]. Taken together, the fact that paraformaldehyde fixation essentially eliminates receptor lateral diffusion along with the fact that the TACF decays do not differ significantly from those produced from the IgE-only treatment group suggests that such a complication is not present. This is because such a complication, if it existed, would be present in the IgE-only treatment group, the contribution to paraformaldehyde TACF decays would be negligible, and this difference would be apparent when comparing the paraformaldehyde treatment group to the IgE treatment group.

Furthermore, theoretical considerations suggest that such a complication resulting from

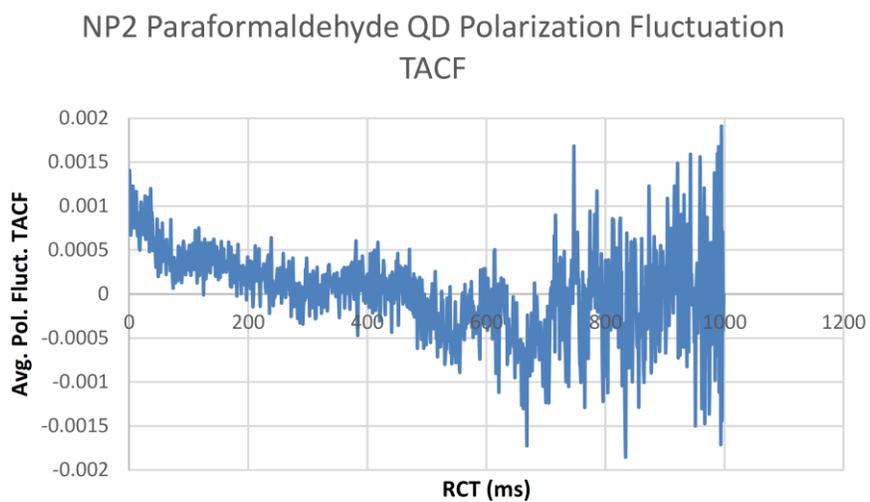
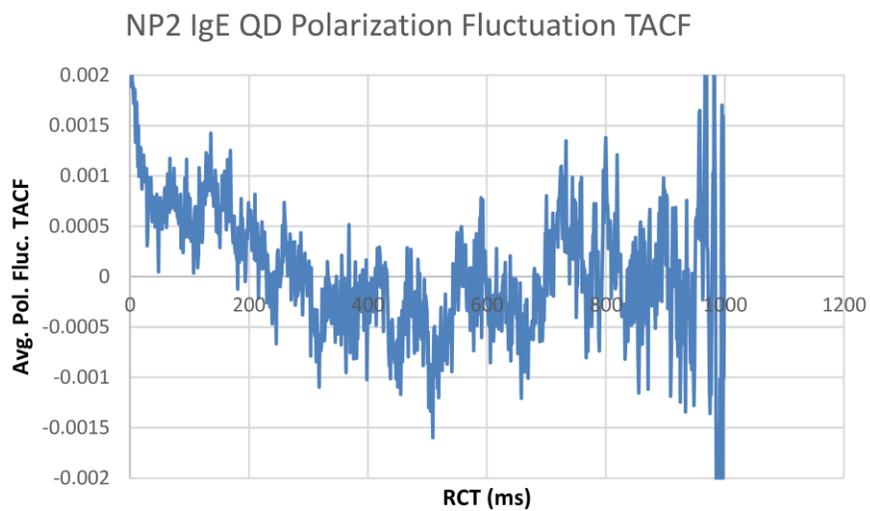


Figure 2.13: Polarization fluctuation TACF for a single QD from the NP2-IgE subgroup (top panel) and from the NP2-Pf subgroup (bottom panel). The similarity between the two traces provides an argument against a lateral diffusion distortion of these traces.

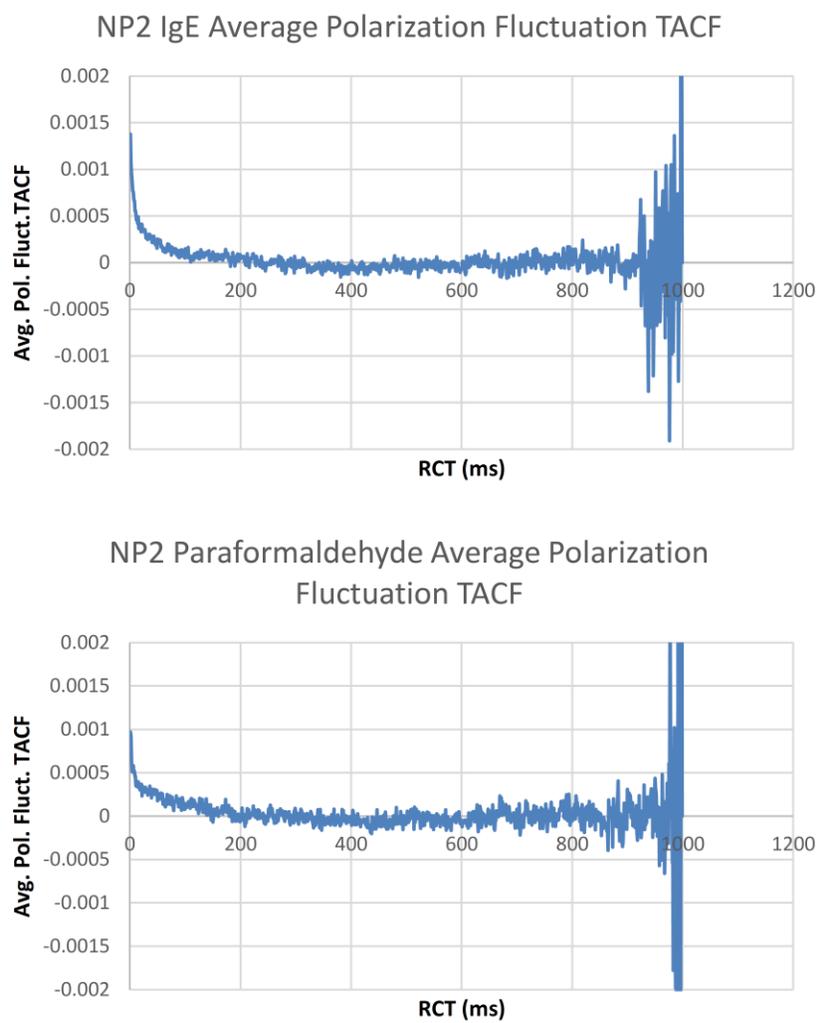


Figure 2.14: Average polarization fluctuation TACF for NP2-IgE subgroup (top panel) and for the NP2-Pf subgroup (bottom). The similarity between the two traces provides an argument against a lateral diffusion artifact distorting of these traces.

lateral diffusion of receptors will not adversely affect our measurements because the fluctuations caused, if present, would occur on a faster timescale than those of the rotational correlation times obtained. Consider the conservative, worst-case scenario in which the offset of the two images Δx is as great as 10 nm and the diffusion constant D is as small as $5 \times 10^{-11} \text{ cm}^2 \text{ s}^{-1}$ [31]. Since the characteristic lateral diffusion time is $\Delta x^2 / 4D$, this would give, at longest, a characteristic diffusion time of 5 ms. This is a worst case scenario and this is still a much faster timescale than our measured rotational correlation times. Thus the timescale of such lateral diffusion-induced fluctuations is likely shorter than could be reasonably expected to interfere with our measurements.

2.4.3. Reproducibility of Data Collected using Non-Polarized Excitation

One concern is whether our measurements would be reproducible. Accordingly, an important question we sought to resolve was whether, for a given treatment, there would be a statistically significant difference between the two non-polarized groups, NP1 and NP2. Because these two groups both consisted of dots excited by non-polarized light with data collected in an identical manner, we expected that there would not be a statistically significant difference between them. If this was found to be the case, this would be an indication that our experiments were reproducible.

To answer this question, we compared both the weighted geometric averages of the rotational correlation times and weighted linear averages of the $G_p(0)$ s as well as the histogram distributions for these two groups. As an example, consider the NP1 and NP2 groups treated with

cytochalasin D (see Tables 2.2 and 2.3). Initial inspection of the data processed by the final primary and secondary Mathematica analysis programs suggests that the averages of the rotational correlation times for these two groups are similar (105 ms with a standard error of ± 104 ms for N=34 selected QDs vs 97 ms with a standard error of ± 16 ms with N=21). A more statistically rigorous analysis employing a two-tailed t-test with $\alpha=0.05$ gives a t-value for the difference between these rotational correlation times of 0.078 which is substantially less than the critical value of 2.086 obtained for 20 degrees of freedom. Regarding the observed $G_p(0)$ s, initial examination suggests these are also very similar for these two groups ((1.2 with a standard error of ± 0.2) $\times 10^{-3}$ vs (1.3 with a standard error of ± 0.2) $\times 10^{-3}$ respectively). A two-tailed t-test with $\alpha=0.05$ gives a t-value for the difference between these amplitudes of which is -0.301 which is also significantly smaller than the critical value of -2.086. These tests suggest that we can accept the null hypothesis which states that these values are essentially the same and conclude that there is no significant difference between the NP1 and NP2 groups. The averages for the rotational correlation times and $G_p(0)$ s for the other four treatments also support this conclusion. The histograms of both the rotational correlation times and amplitudes corroborate this point as in neither case does it appear that the histogram for the NP1 group is shifted significantly one way or the other relative to the NP2 group and this seems to be the case for the other four treatments as well (see Figures 2.15 and 2.16). Such an affirmation lends credence to the reproducibility of our measurements. Taking this conclusion into consideration, it was determined that it would be appropriate to merge both the averages and histograms for these groups and use these combined results for further analyses.

Table 2.2: Comparison of rotational correlation times and associated statistics for QD bound to cells in the two non-polarized subgroups (NP1 and NP2) in each of the six treatment groups. Measurements for dry dots were only conducted with non-polarized excitation (NP2-dry).

Group	1st reagent	2nd reagent	Type of Excitation	Wtd Geo. Avg. RCT (ms)	SD RCT (ms)	Std. Err. RCT (ms)	Tot. QD	QD Selected
NP2-dry	-	-	Non-Pol.	8.0	9.3	2.3	106	17
NP1-IgE	IgE	-	Non-Pol.	104.1	295.9	45.7	86	42
NP2-IgE	IgE	-	Non-Pol.	74.4	147.8	18.3	339	65
NP1-DNP-BSA	IgE	DNP-BSA	Non-Pol.	226.7	544.9	73.5	93	55
NP2-DNP-BSA	IgE	DNP-BSA	Non-Pol.	105.0	200.3	34.4	140	34
NP1-Pf	IgE	Pf	Non-Pol.	123.9	488.2	86.3	73	32
NP2-Pf	IgE	Pf	Non-Pol.	203.0	497.4	81.8	151	37
NP1-M β CD	IgE	M β CD	Non-Pol.	57.5	81.1	22.5	44	13
NP2-M β CD	IgE	M β CD	Non-Pol.	89.0	57.6	12.6	68	21
NP1-CytoD	IgE	CytoD	Non-Pol.	104.7	606.8	104.1	82	34
NP2-CytoD	IgE	CytoD	Non-Pol.	96.6	72.8	15.9	112	21

Table 2.3: Comparison of polarization TACF amplitudes and associated statistics for QD bound to cells in the two non-polarized subgroups (NP1 and NP2) for each of the six treatment groups. Measurements for dry dots were only conducted with non-polarized excitation (NP2-dry).

Group	1st reagent	2nd reagent	Excitation	Wtd. Arith. Avg. $G_{pp}(0)$	SD $G_{pp}(0)$	SE $G_{pp}(0)$	Tot. QD	QD Selected
NP2-dry	-	-	Non-Pol.	0.00664	0.00392	0.00095	106	17
NP1-IgE	IgE	-	Non-Pol.	0.00165	0.00133	0.00021	86	42
NP2-IgE	IgE	-	Non-Pol.	0.00164	0.00133	0.00017	339	65
NP1-DNP-BSA	IgE	DNP-BSA	Non-Pol.	0.00133	0.00092	0.00012	93	55
NP2-DNP-BSA	IgE	DNP-BSA	Non-Pol.	0.00144	0.00109	0.00019	140	34
NP1-Pf	IgE	Pf	Non-Pol.	0.00140	0.00123	0.00022	73	32
NP2-Pf	IgE	Pf	Non-Pol.	0.00134	0.00080	0.00013	151	37
NP1-M β CD	IgE	M β CD	Non-Pol.	0.00124	0.00165	0.00046	44	13
NP2-M β CD	IgE	M β CD	Non-Pol.	0.00116	0.00078	0.00017	68	21
NP1-CytoD	IgE	CytoD	Non-Pol.	0.00125	0.00125	0.00021	82	34
NP2-CytoD	IgE	CytoD	Non-Pol.	0.00133	0.00082	0.00018	112	21

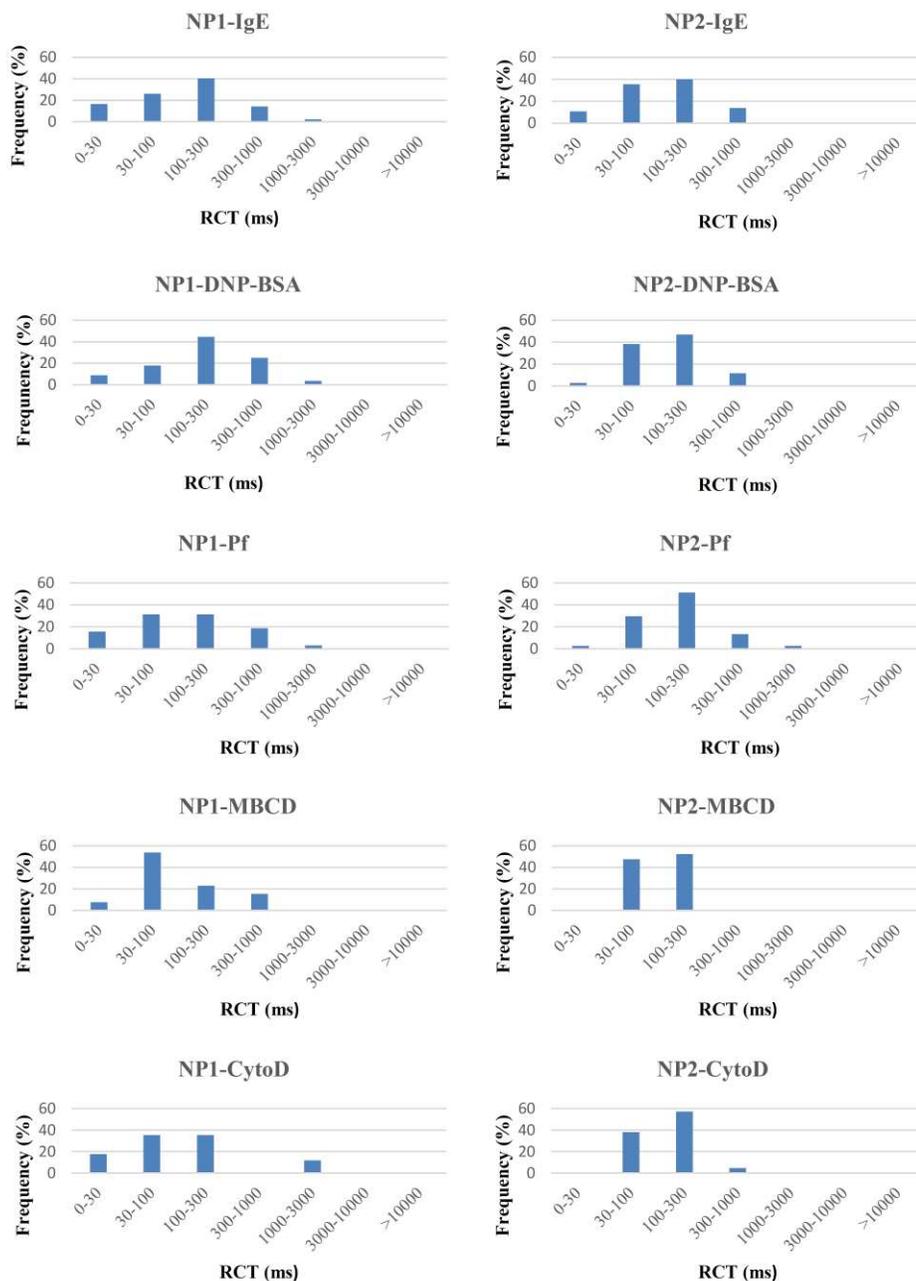


Figure 2.15: Comparison of the histogram distributions of rotational correlation times of receptor-bound QD on cells from the two subgroups for which experiments were carried out with non-polarized excitation (NP1 and NP2) for each of the six treatment groups.

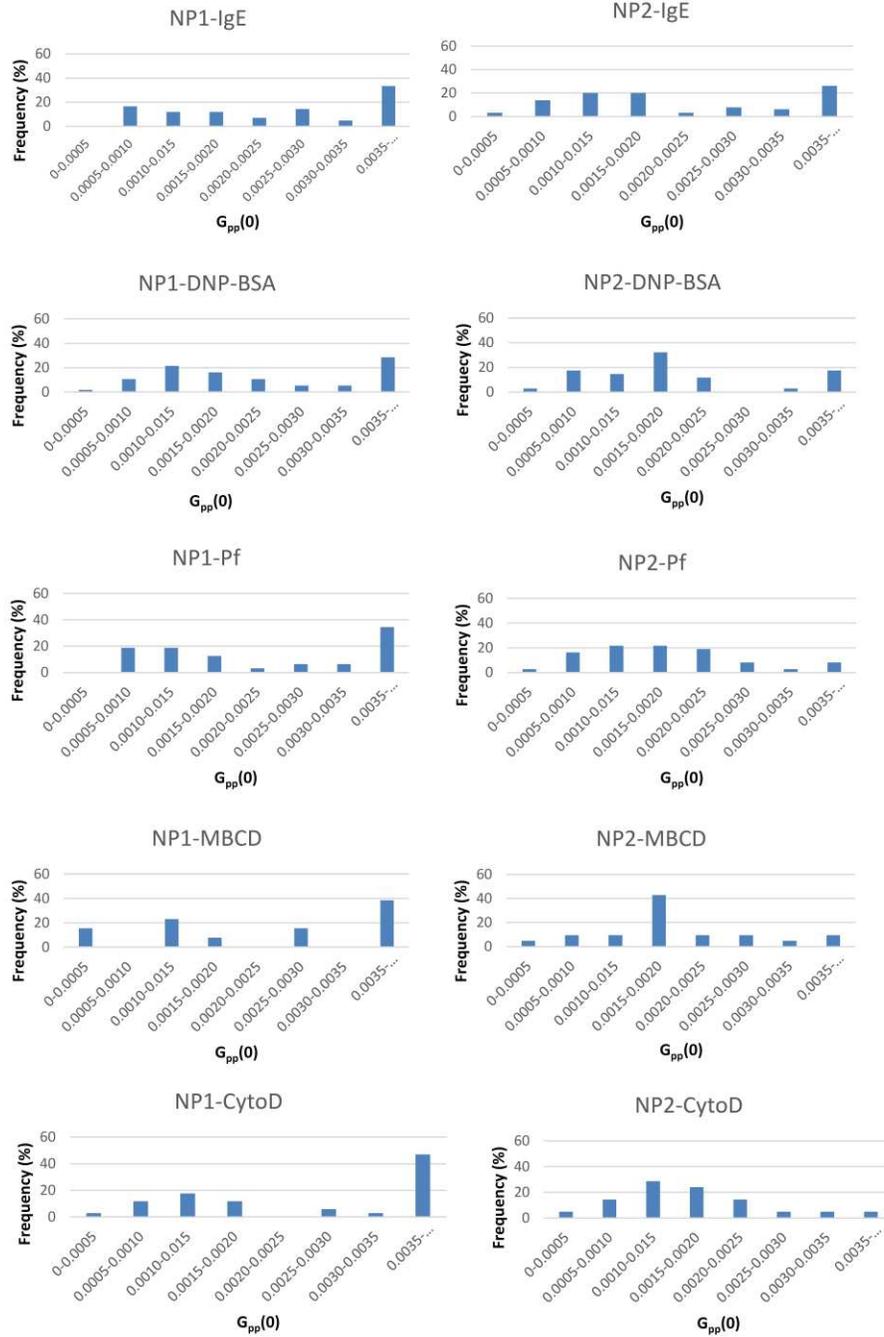


Figure 2.16: Comparison of polarization fluctuation TACF amplitude histogram distributions for QD bound to cells from of the two non-polarized subgroups (NP1 and NP2) for the each of the six treatment groups.

2.4.4. Comparison of polarized to non-polarized groups

Another objective of these analyses was to determine whether there would be a difference between results obtained for experiments in which samples were excited with polarized light and non-polarized light, respectively. If we use a convention in which the longitudinal/optical axis is designated as the z-axis, then the molecule can be excited in any direction within the xy-plane. This could be the x-axis, y-axis, or a combination in which the vector could have components of each. In the experiments in which we used polarized light to excite the sample, we chose to use excitation light polarized horizontally/along the x-axis. In others, we used non-polarized light. Regardless of whether the molecule is excited by polarized or non-polarized light, it can only be excited by light in an orientation that is in alignment with its absorption dipole. The fluorescence produced by non-polarized excitation can, in essence, be considered akin to the fluorescence that would be obtained if the sample was excited in the x- and y-polarized orientations *separately*. Intuitively, non-polarized excitation gives ensemble and average single-molecule polarizations equal to zero. The reason for this is that, in this scenario, the average fluorescence intensities are equal in the two orientations (*i.e.* $\langle I_x \rangle = \langle I_y \rangle$) and therefore $\langle r_{\text{Ens}} \rangle = 0$. However, less obvious is that the polarization fluctuation TACF amplitude in this scenario does not equal zero. Although it might also be expected intuitively that polarized excitation should give a larger $G_p(0)$, theoretical considerations suggest this is not the case. As observed in Table 2.1 and explained in the accompanying discussion (*vide supra*), the difference between the initial amplitude and limiting value of the polarization fluctuation TACF, $G_p(0) - G_p(\infty)$ should be equivalent for non-polarized and polarized excitation.

Examination of Table 2.4 reveals no consistent difference in the geometric averages of the rotational correlation times between the polarized (P3) and the merged non-polarized (NP1 and NP2) subgroups. For some treatment groups, polarized excitation gives a larger rotational correlation time and for others, it is smaller. Examination of Table 2.5 shows arithmetic averages of polarization fluctuation TACF amplitudes that are very similar. As an example, consider the P3 and merged NP1 and NP2 IgE subgroups. The P3 IgE subgroup had an average amplitude of 0.00164 ± 0.00133 and the NP1 and NP2 merged subgroups had an average of 0.00163 ± 0.00103 . The t-value for the difference between these two sample means is 0.097 which is less than the critical value of ~ 1.992 for $\alpha=0.05$ and 74 degrees of freedom. This suggests we should not reject the null hypothesis which suggests that the population means are the same. A similar conclusion could be drawn when comparing the other polarized and non-polarized subgroups. Together, the lack of a consistent difference in the rotational correlation times and the similarity of the values for the amplitudes suggest that experiments conducted with polarized and non-polarized excitation do not yield significantly different results. In agreement with this finding, the histograms in Figure 2.17 and Figure 2.18 do not indicate consistent, significant differences in the respective rotational correlation times and amplitude distributions. No trend suggesting that polarized excitation gives a higher amplitude than non-polarized can be observed in Table 2.5 or Figure 2.18. Thus we concluded that these observations provide evidence that experiments conducted with polarized and non-polarized excitation are equivalent in terms of the data they produce.

2.4.5. Effects of cell treatments on QD rotation as observed by imaging

Table 2.4: Comparison of rotational correlation times and associated statistics of polarized (P3) and merged non-polarized (NP1 and NP2) subgroups for each of the six treatment groups. Weighted averages were calculated for the rotational correlation times and associated statistics when merging the NP1 and NP2 groups.

Group	1st reagent	2nd reagent	Excitation	Wtd Geo. Avg. RCT (ms)	SD RCT (ms)	Std. Err. RCT (ms)	Tot. QD	QD Sel.
NP2-dry	-	-	Non-Pol.	8.0	9.3	2.3	106	17
NP1 and NP2-IgE	IgE	-	Non-Pol.	86.0	205.9	19.9	425	107
P3-IgE	IgE	-	Pol.	208.3	573.8	66.3	340	75
NP1- and NP2-DNP-BSA	IgE	DNP-BSA	Non-Pol.	180.2	413.3	43.8	233	89
P3-DNP-BSA	IgE	DNP-BSA	Pol.	88.7	146.7	39.2	189	14
NP1- and NP2-Pf	IgE	Pf	Non-Pol.	166.3	493.2	59.4	224	69
P3-Pf	IgE	Pf	Pol.	108.8	180.6	35.4	187	26
NP1- and NP2-M β CD	IgE	M β CD	Non-Pol.	76.9	66.6	11.4	112	34
P3-M β CD	IgE	M β CD	Pol.	145.0	108.3	34.3	124	10
NP1- and NP2-CytoD	IgE	CytoD	Non-Pol.	101.6	402.9	54.3	194	55
P3-CytoD	IgE	CytoD	Pol.	135.2	273.6	59.7	163	21

Table 2.5: Comparison of polarization TACF amplitudes and associated statistics of merged non-polarized subgroups (NP1 and NP2) and polarized subgroup (P3) for each of the six treatment groups.

Group	1st reagent	2nd reagent	Excitation	Wtd. Arith. Avg. $g_{pp}(0)$	SD $g_{pp}(0)$	SE $g_{pp}(0)$	Tot. QD	QD Sel.
NP2-dry	-	-	Non-Polarized	0.00664	0.00392	0.00095	106	17
NP1- and NP2-IgE merged	IgE	-	Non-Polarized	0.00164	0.00133	0.00013	425	107
P3-IgE	IgE	-	Polarized	0.00163	0.00103	0.00012	340	75
NP1 and NP2-DNP-BSA merged	IgE	DNP-BSA	Non-Polarized	0.00137	0.00098	0.00010	233	89
P3-DNP-BSA	IgE	DNP-BSA	Polarized	0.00150	0.00121	0.00032	189	14
NP1- and NP2-Pf merged	IgE	Pf	Non-Polarized	0.00137	0.00100	0.00012	224	69
P3-Pf	IgE	Pf	Polarized	0.00128	0.00068	0.00013	187	26
NP1 and NP2-M β CD merged	IgE	M β CD	Non-Polarized	0.00119	0.00111	0.00019	112	34
P3-M β CD	IgE	M β CD	Polarized	0.00136	0.00054	0.00017	124	10
NP1- and NP2-CytoD merged	IgE	CytoD	Non-Polarized	0.00128	0.00108	0.00015	194	55
P3-CytoD	IgE	CytoD	Polarized	0.00180	0.00184	0.00040	163	21

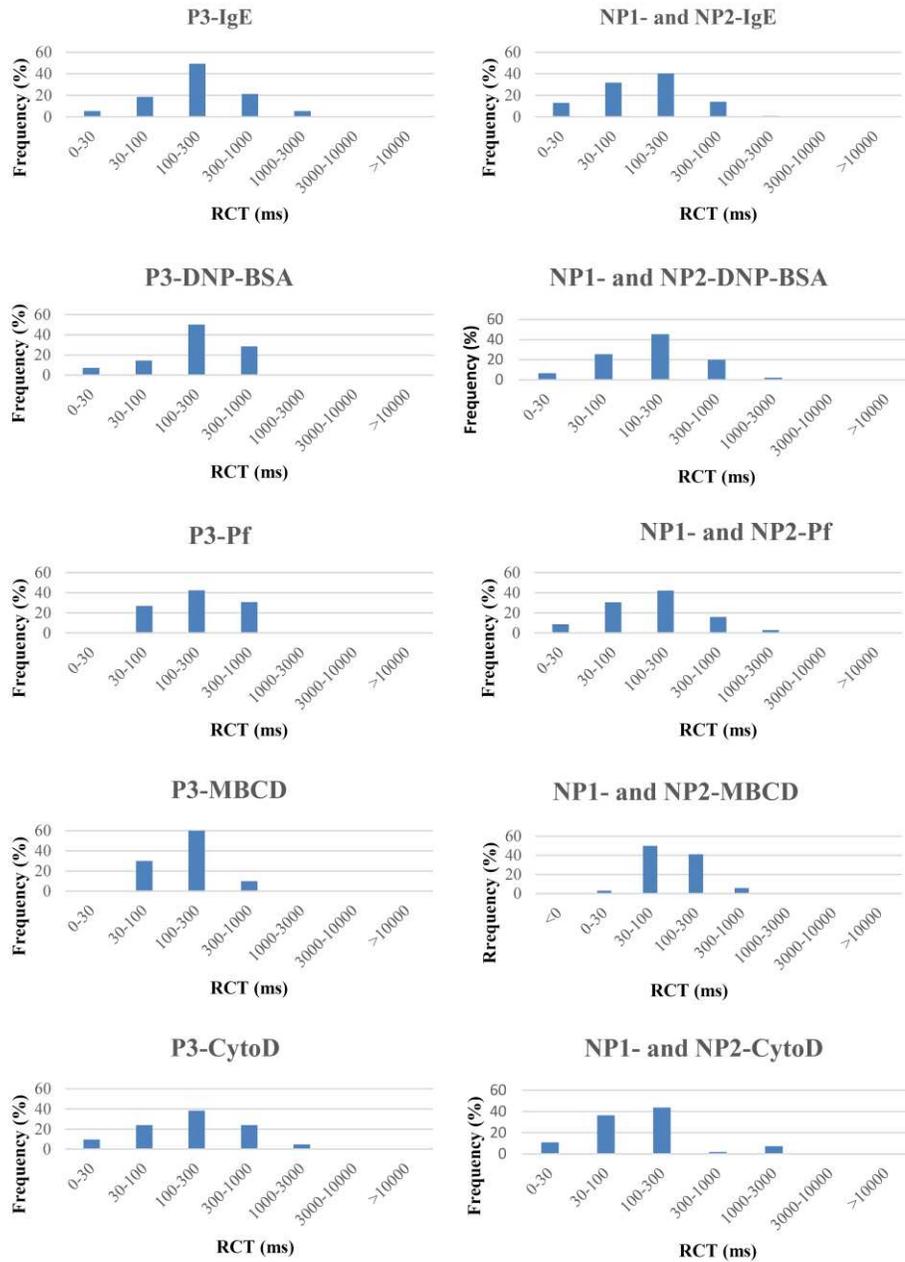


Figure 2.17: Comparison of histogram distributions of rotational correlation times of QD bound to cells from subgroups for which experiments were conducted with polarized excitation (P3) and merged results for subgroups in which experiments were carried out with non-polarized excitation (NP1 and NP2) for each of the six treatment groups.

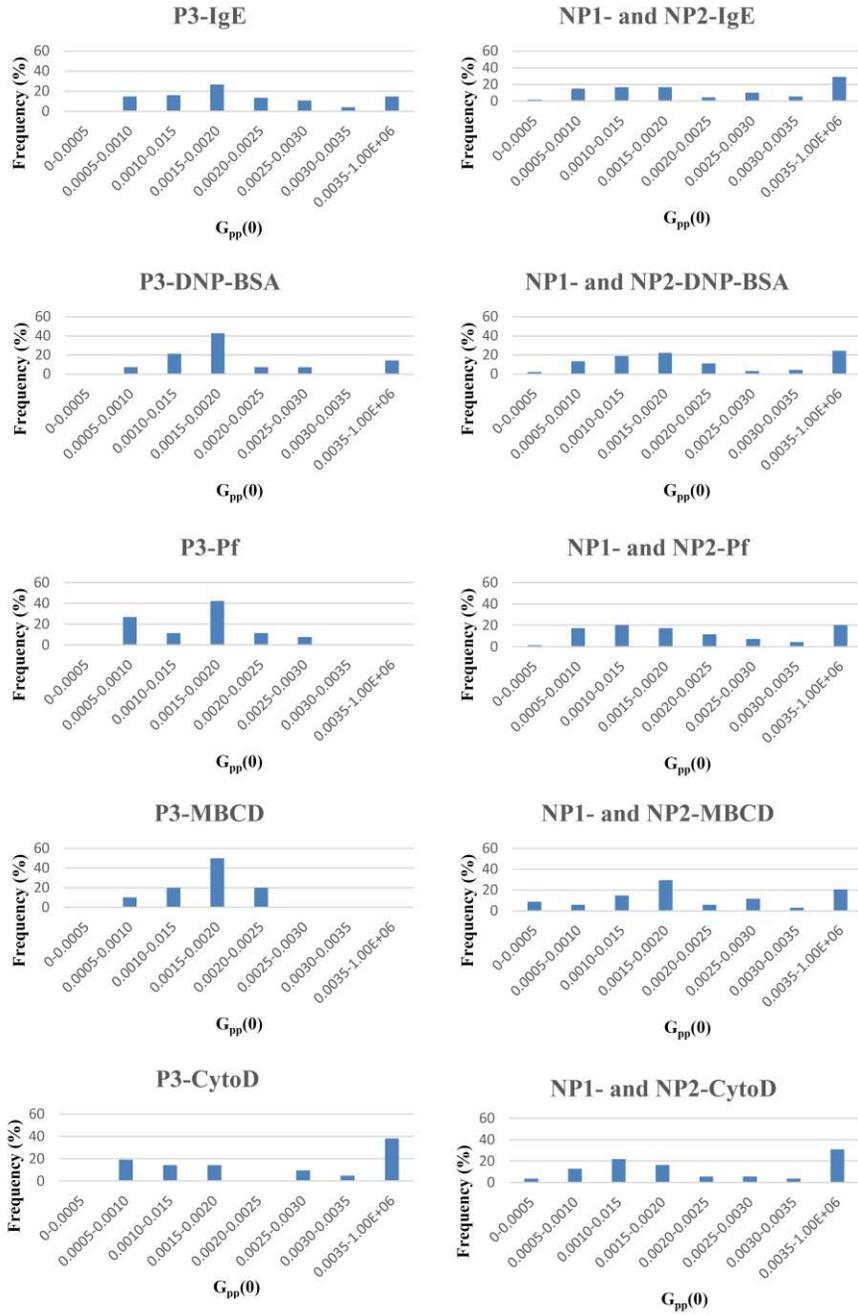


Figure 2.18: Comparison of polarization fluctuation TACF amplitude histogram distributions for QD bound to cells from the polarized (P3) and merged non-polarized (NP1 and NP2) subgroups for the six treatment groups.

A major aim of this study was to determine what differences in rotational behavior, if any, might result from treatment of cells with various agents expected to perturb receptor rotation either by changing receptor interactions or altering the local environment of the receptor. Figure 2.19 and Table 2.6 show histogram distributions of rotational correlation times and accompanying statistics respectively for the six different treatments. Regarding rotational correlation times, it is apparent from the histograms that the range of rotational correlation times exhibited spans 2-3 orders of magnitude (~10 ms to ~2000 ms). As previously mentioned, this is why we have calculated geometric averages instead of arithmetic averages for the rotational correlation times. If we had calculated arithmetic averages, QDs exhibiting slow rotation would have had a disproportionate effect on the average, increasing it substantially. With the obvious exception of dry QDs which are immobile, averages for rotational correlation times ranged from about 100-150 ms. With the exception of the M β CD-treated group and the dry QDs for which the standard deviation scaled with the lower average, the standard deviations of the rotational correlation times ranged from ~ \pm 350-400 ms. However, because the groups differed in the number of QDs selected for analysis, it was more informative to calculate the standard error to account for differences in sample size. These values for the most part ranged from ~ \pm 25-40 ms.

Figure 2.20 and Table 2.7 show histogram distributions of $G_p(0)$ and accompanying statistics respectively for the six different treatments. From this data, a few general observations can be made. Firstly, it is apparent that, with the exception of the group treated with only IgE, which has an amplitude of 0.00164, the amplitudes are similar in magnitude. The IgE-treated group has a slightly larger average $G_p(0)$. If this does indeed represent a statistically significant difference, one explanation for this could be because these other treatments which include the

Table 2.6: Comparison of rotational correlation times and associated statistics of all merged subgroups (NP1, NP2, and P3) for the six treatment groups. Weighted averages were calculated for the rotational correlation times and associated statistics when merging the NP1, NP2, and P3 groups.

Group	1st reagent	2nd reagent	Wtd Geo. Avg. RCT (ms)	SD RCT (ms)	Std. Err. RCT (ms)	Tot. QD	QD Selected
NP2-dry	-	-	8.0	9.3	2.3	106	17
NP1, NP2, and P3 IgE merged	IgE	-	136.4	357.5	26.5	765	182
NP1, NP2, and P3 DNP-BSA merged	IgE	DNP-BSA	167.8	377.0	37.2	422	103
NP1, NP2, and P3 Pf merged	IgE	Pf	150.5	407.6	41.8	411	95
NP1, NP2, and P3 M β CD merged	IgE	M β CD	92.4	76.1	11.5	236	44
NP1, NP2, and P3 CytoD merged	IgE	CytoD	110.9	367.1	42.1	357	76

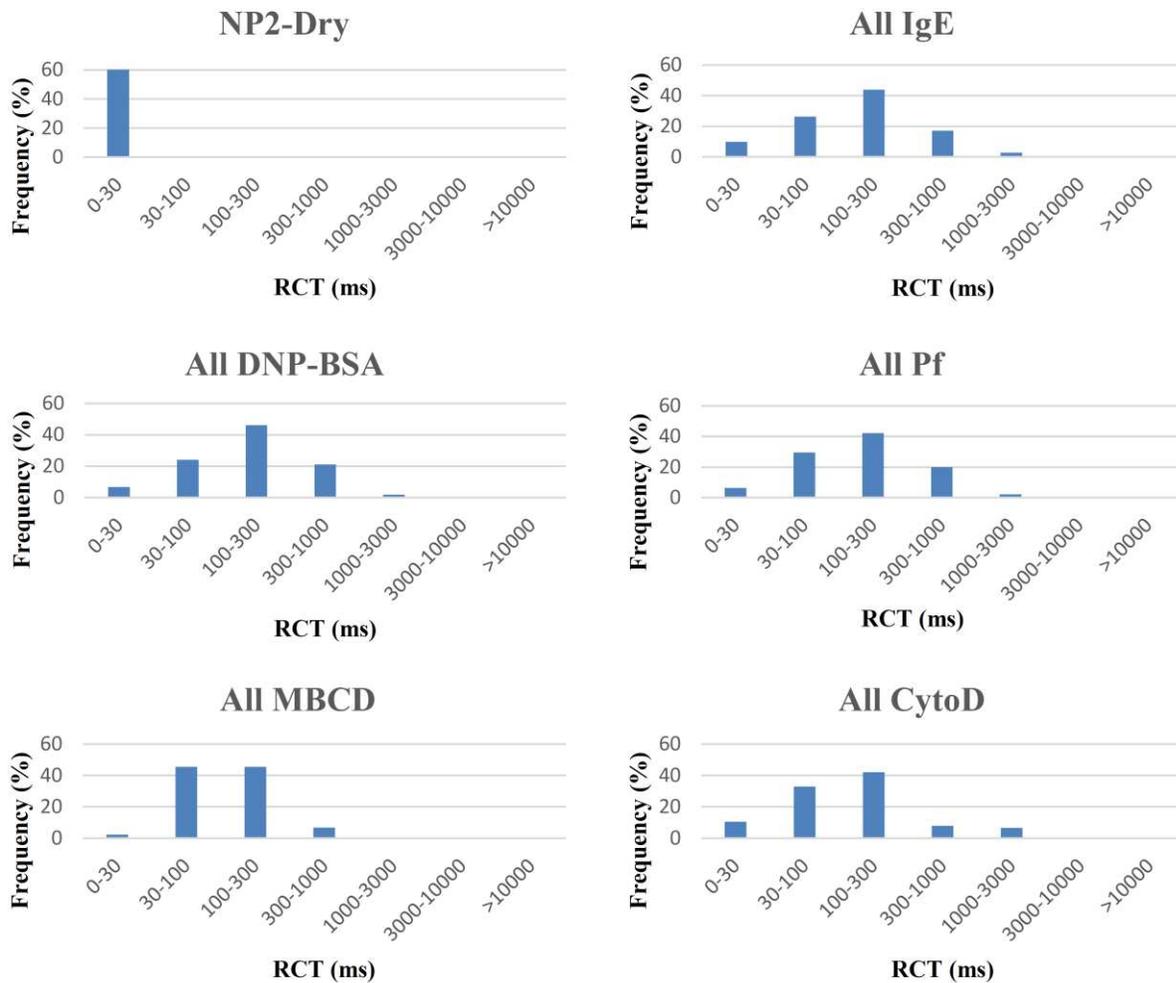


Figure 2.19: Comparison of histograms distributions of rotational correlation times of all merged subgroups including those in which experiments were conducted with non-polarized (NP1 and NP2) and polarized excitation (P3) for each of the six different treatment groups. The distributions for the three subgroups for each of the treatment groups have been summed to give the merged histograms displayed above.

Table 2.7: Comparison of polarization TACF amplitudes and associated statistics of all merged subgroups (NP1, NP2, and P3) for each of the six treatment groups.

Group	1st reagent	2nd reagent	Wtd. Arith. Avg. $G_{pp}(0)$	SD $G_{pp}(0)$	SE $G_{pp}(0)$	Tot. QD	QD Selected
NP2-dry	-	-	0.00664	0.00392	0.00095	106	17
NP1, NP2, and P3 IgE merged	IgE	-	0.00164	0.00121	0.00009	765	182
NP1, NP2, and P3 DNP-BSA merged	IgE	DNP-BSA	0.00139	0.00101	0.00010	422	103
NP1, NP2, and P3 Pf merged	IgE	Pf	0.00135	0.00091	0.00009	411	95
NP1, NP2, and P3 M β CD merged	IgE	M β CD	0.00123	0.00098	0.00015	236	44
NP1, NP2, and P3 CytoD merged	IgE	CytoD	0.00143	0.00129	0.00015	357	76

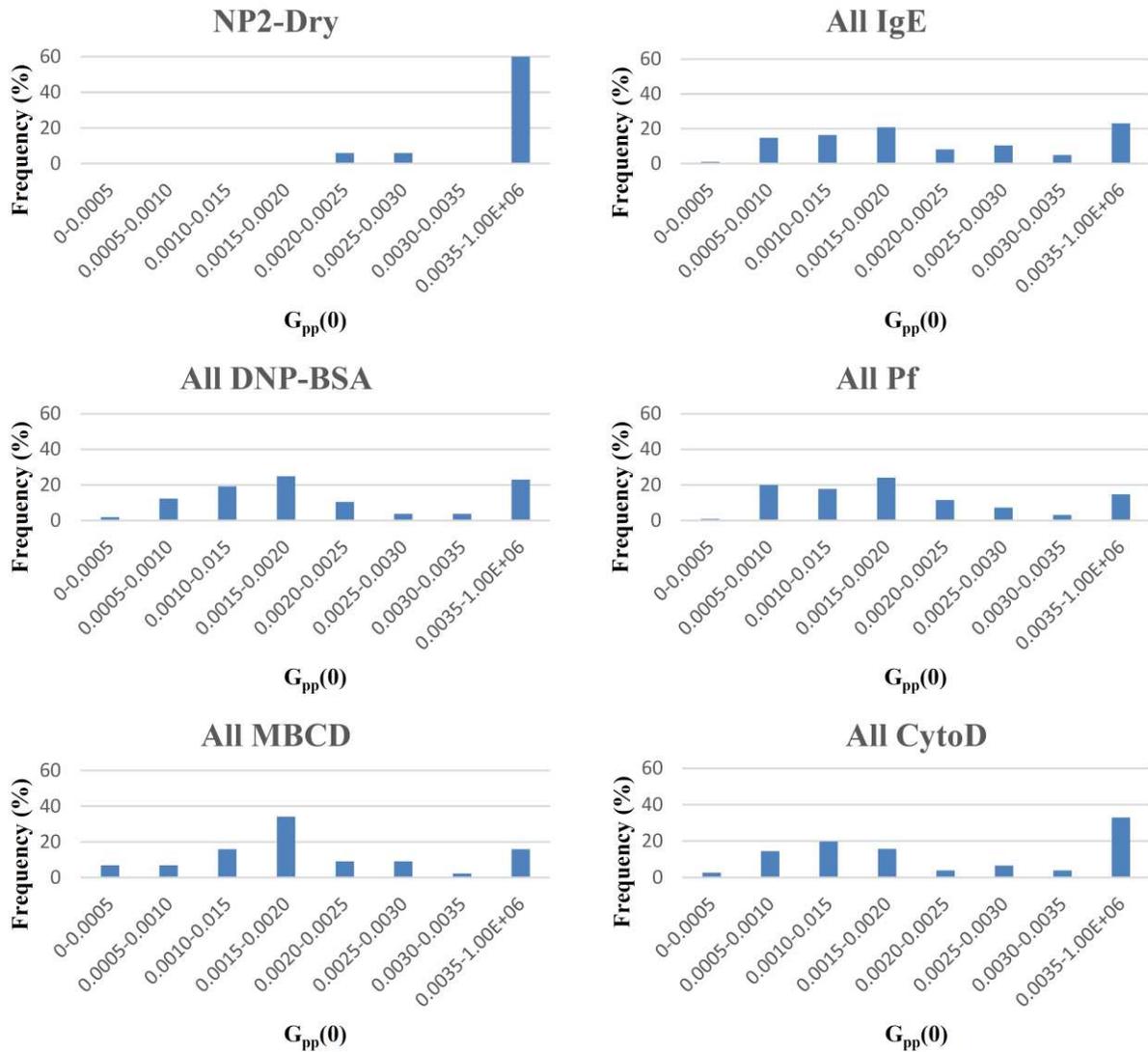


Figure 2.20: Comparison of histogram frequency distributions of polarization fluctuation TACF amplitudes for each of the six treatment groups. The distributions for the three subgroups for each of the treatment groups have been summed to give the merged histograms displayed above. Since the dry treatment group only had one subgroup taken with non-polarized excitation (NP2), this group is displayed by itself.

DNP-BSA, paraformaldehyde, M β CD, and CytoD -treated groups, immobilize a fraction of receptors. If these immobile receptors do not contribute to the polarization fluctuation TACF amplitude, this could potentially be the reason behind the reduced amplitudes for these treatment groups relative to the IgE-only group. With the exception of the CytoD-treated group, the standard deviations of the $G_p(0)$ s for these groups are also reduced when compared to the IgE-only treated group, which has a standard deviation of 0.00121. The CytoD group has a very similar standard deviation of 0.00129. These lowered standard deviations could also be indicative of receptor immobilization induced by these treatments, as this would also be expected to decrease the breadth of the distribution of amplitudes exhibited by the receptors. The standard errors, however, are all very similar at ~ 0.0001 . The M β CD and CytoD groups have slightly elevated standard errors of 0.000148. What follows is a more in-depth discussion of the effects of each treatment.

2.4.5.1. Dry QDs

Because dry QDs are obviously expected to be immobile, we hypothesize their polarization correlation curves should decay to 0 almost immediately. The exception is the first point which must be non-zero due to correlation of shot noise and detector after-pulsing in the correlation of each point with itself. Regardless of the size of the amplitude $G_p(0)$ by the second point all of the polarization should have decayed to 0. In this way a fast rotational correlation time can be an indicator of immobility.

For the dry QDs we employed as a control, after processing with Image Corr v.140 and MergeFiles v.39 programs, raw data output from the full set of QDs indicated that 28 out of 106 QDs exhibited rotational correlation times $<1 \mu\text{s}$ for a total of 26% of QDs exhibiting very fast correlation times. A higher proportion of QDs were represented in this category than for other treatment categories as was expected. Final data output after rejection of QDs based on various criteria including the rotational correlation time τ_D and the amplitude $G_{pp}(0)$ and several other criteria as well as manual rejection of QDs with unreasonable parameters left 17 QDs all exhibiting correlation times in the 0-30 ms bin of our histogram (see Figure 2.19). Selection criteria were the same as for other treatment groups with the exception of the rotational correlation time which was constrained to 0-3 μs instead of 1-200 μs and the amplitude for which the range was wider to allow for the inclusion of QDs with large amplitudes. As can be seen in Figure 2.20, the overwhelming majority of these fell in the top bin of the histogram with values between 0.0035 and 1×10^6 . Though this bias towards short correlation times and long correlation times could be considered an artifact of the aforementioned selection criteria with the change in the range permitted for the rotational correlation time, it is worth mentioning again that the raw results from the program before undesired QD outliers were removed did give a significant portion of QDs (larger than other treatment groups) with very fast correlation times. These results are consistent with the previously explained expectations for these dry QDs.

2.4.5.2. IgE only

For the cells treated with IgE only, we would initially expect to see rotational correlation times corresponding to the hydrodynamic rotation of the QD-labeled IgE-receptor complex.

Since we have labeled the ligand to which the receptor binds instead of directly labeling the receptor, we might initially expect a longer rotational correlation time than we would obtain for the unbound receptor. The IgE-bound FcεRI complex has a total molecular mass of ~250 kDa which is significantly larger than the lone receptor (which has a molecular mass of ~100 kDa). However, considering the IgE is located extracellularly, we would not expect a substantial difference in the rotational correlation time between the bound and unbound receptor because the receptor itself, being confined to the membrane, is thus responsible for the majority of the resistance to rotational motion and the IgE, despite its significant size, would slow this motion to a negligible extent. Our group has previously studied the rotation of this system by time-resolved phosphorescence anisotropy and determined the rotational correlation time of the erythrosin-labeled IgE-FcεRI to be $79 \pm 31 \mu\text{s}$ at 4°C [32]. In close agreement with these results, our group has also studied this system using fluorescence depletion anisotropy and determined the rotational correlation time of the eosin isothiocyanate-conjugated IgE-FcεRI complex to be $79 \pm 4 \mu\text{s}$ [33]. In the absence of polyvalent antigen or other receptor-crosslinking agents, we would also not expect a distribution of receptor complex sizes and corresponding rotational correlation times, but rather a single rotational correlation time with a value $\sim 80 \mu\text{s}$ corresponding to the rotation of this receptor-ligand complex.

Our imaging method yielded a weighted geometric average rotational correlation time of 136 ms at room temperature (25°C) with a standard error of $\pm 27 \text{ ms}$ (see Table 2.6). This value is several orders of magnitude larger than what could be attributed to the hydrodynamic rotation of the receptor. In fact, our imaging method is unable to detect such fast rotation as this is below the 10 ms minimum frame time of our present camera. Also, the standard deviation and

histogram (see Figure 11) indicates that the variability in rotational correlation time from dot to dot is greater than might be expected. The reason we observe such a long rotational correlation time may be because what we are observing is not in fact the hydrodynamic rotation of the receptor, but instead the rotation or libration of domains containing the receptor such as lipid rafts which could plausibly exhibit rotation on this timescale. This would also provide an explanation for the variability, as we would expect several distinct receptor populations occupying domains of varying sizes. The polarization TACF amplitude $G_p(0)$ was 0.001637 ± 0.001205 (see Table 2.7) which is slightly higher than the other treatments and the histogram for these amplitudes (see Figure 2.20) appears to be shifted slightly to the right as well. The reason for this will be considered in our discussion of the remaining treatments.

2.4.5.3. DNP-BSA and IgE

DNP-BSA is a ~66 kDa hapten-carrier adduct [34] that consists of bovine serum albumin conjugated with multiple dinitrophenyl haptens. According to the manufacturer ThermoFisher Scientific, typically this is prepared with ~25 of these per protein [35]. We used a previously-published protocol [36] to derivatize DNP-BSA which give an average of ~11-14 DNP-groups per molecule as performed previously [31, 32]. These dinitrophenyl haptens have specificity for A2 IgE antibody and therefore this agent can, by virtue of its multi-valency, crosslink multiple IgE-bound Fc ϵ RI receptors, thereby slowing their rotation. Each Fab arm of the IgE ligand is capable of binding a dinitrophenyl group of the haptened BSA and thus this ligand is bivalent. This allows for an even greater degree of crosslinking creating even bigger complexes involving multiple DNP-BSA proteins and correspondingly longer rotational correlation times. Such

slowed rotational behavior has been observed experimentally in time-resolved phosphorescence anisotropy measurements in which it was found that such treatment led to an increase in the limiting anisotropy of the receptor which suggested rotational immobilization of a larger fraction of receptors on the 100 μ s timescale [27]. It is for this reason that we expect pre-treatment with IgE followed by treatment with DNP-BSA should slow protein rotation relative to when cells are treated with solely IgE. We expect the retardation of rotational diffusion to occur to a significant degree given the results from the time-resolved phosphorescence anisotropy experiments. For the rotational correlation time to have remained relatively constant on these timescales (reflecting receptors that did not undergo crosslinking) while the limiting anisotropy increased (reflecting a larger population of receptors that are immobile on this timescale), the rotational correlation time must have increased at least 10-fold upon crosslinking into larger complexes. Otherwise the increase in size would have been detectable by this method as a curve shifted to the right with a longer rotational correlation time.

It is difficult to predict the exact extent of binding that will occur and resultant rotational correlation times we should expect to observe. For example, if we assumed an average of 14 DNP groups per protein with two DNPs bound per IgE, this would give 7 IgE-receptor complexes crosslinked by a single molecule of BSA. However, the situation is considerably more complex than this. Firstly, this assumes that all the DNPs on BSA bind to an IgE and that both sites of every IgE molecule are occupied. In reality, such complete saturation is unlikely especially considering some binding sites may be inaccessible (*e.g.* buried within the BSA protein or on the side facing away from the membrane). Another issue with such a simplified model is that it is possible for an IgE molecule, because it has two Fab fragments and is therefore

divalent, to bind two different molecules of DNP-BSA and thus form a crosslinked network of IgE crosslinked with not just a single but multiple molecules of DNP-BSA. It is difficult to predict which kind of binding is more likely. On one hand, if IgE is already bound at one site to a DNP on a BSA protein, then this would put it in closer proximity to other DNP groups on this protein and perhaps make binding more likely. On the other hand, for binding of both sites of a single molecule of IgE to two DNPs on the same molecule of BSA to occur, a very particular spacing of the DNPs on the BSA would be required. Steric considerations are necessary in order to determine the most likely arrangements and sizes of such a crosslinked complex. In any case, on the millisecond timescale as examined by our imaging method, we would expect these complexes to exhibit rotational correlation times lower than what we might expect from the lipid domains and so we cannot rule out the possibility that we may counterintuitively observe a lower average rotational correlation time than we did for IgE despite the fact that we expect larger receptor complexes upon treatment with DNP-BSA.

We would also expect the DNP-BSA-treated group to display more variability in rotational correlation times from dot-to-dot because we expect a distribution of complex sizes. As previously mentioned, assuming 14 DNP groups per molecule of BSA, a single molecule of DNP-BSA could crosslink up to 7 receptor-antibody complexes via binding of the IgE Fab fragment to the dinitrophenyl group at which point it would be saturated, but it is reasonable to suspect that a portion of the DNP groups would not undergo binding to an IgE antibody or that some IgE-receptor complexes will not be bound to DNP-BSA. Furthermore, it is not necessary that both Fab fragments of IgE bind to the same molecule of BSA, and therefore it is possible to have networks of receptors crosslinked with BSA with multiple DNP-BSA molecules. Our

laboratory has calculated the size distributions of receptor-polyvalent ligand aggregates in a different cellular system.

Upon examination of Table 2.6, a rotational correlation time of 168 ms with a standard error of ± 37 ms is observed for the DNP-BSA-treated group. This is slightly higher than what we observed for the IgE-only group, and though the standard error is a little bigger, it is very similar to that seen for the IgE-only treated group. Moreover, the histogram distributions of rotational correlation times appear very similar for these two groups (see Figure 11). Though we would expect larger rotational correlation times for this treatment group than we would for the IgE-only treated group due to extensive crosslinking by DNP-BSA, we also saw rotation occurring on a similar timescale for the IgE-only group which we attributed to the motion of lipid microdomains, and we have every reason to suspect the same phenomenon is occurring in this situation. Thus we once again attributed this rotation to the rafts or other discrete domains containing the receptors, though we do not rule out the rotation of large complexes crosslinked by DNP-BSA especially for those receptors with rotational correlation times in the lowest bins of the histogram. If the fraction of receptors that were observed to exhibit rotational immobility on the 100 μ s timescale in time-resolved phosphorescence anisotropy experiments were not completely immobilized but instead simply exhibited slower rotation than could be observed on this timescale, it is possible that some decay attributable to the rotation of receptors crosslinked by DNP-BSA could be present on a longer timescale in addition to that attributable to the rotation of domains which would be predicted to occur on an even longer timescale. Predicting the range of sizes and corresponding rotational correlation times of such complexes is difficult. Noteworthy is the slightly reduced $G_p(0)$ obtained for the DNP-BSA-treated group (0.00139

± 0.00101) relative to the IgE-only-treated group. One possible explanation for this effect is that treatment with DNP-BSA has resulted in the immobilization of a fraction of receptors, which, because they would no longer contribute to the average, would accordingly result in a decreased average value. This is a plausible explanation, as receptors that are crosslinked and, presumably, immobilized, should exhibit reduced $G_p(0)$ s since the immobilized receptor fraction has $G_p(0)=0$.

2.4.5.4. Paraformaldehyde and IgE

Paraformaldehyde is synthesized via evaporation of aqueous solutions of formaldehyde. Formation involves hydration of formaldehyde followed by nucleophilic addition of the resulting geminal diol to another molecule of formaldehyde forming a hemiacetal or, for subsequent steps in the polymerization, an acetal. It is a linear (non-cyclical) polyoxymethylene with a general molecular formula of $\text{HO}[\text{CH}_2\text{O}]_n\text{H}$ where the number of units n ranges from 8 to 100 with the typical average number of units being 12 [37]. It is a polyacetal in which the terminal carbon positions are classified as hemiacetals and the interior carbon positions as acetals.

Paraformaldehyde solutions that are pH-neutral of ~0.5 wt % are often used as fixatives. The hypothesized mechanism by which paraformaldehyde fixates membrane proteins involves covalent crosslinking of the primary amine $-\text{NH}_2$ groups of lysine residues to $-\text{NH}-$ groups in a peptide backbone [38]. Thus this agent, in contrast to DNP-BSA, the other crosslinking agent we employed, has much less specificity as it can crosslink any membrane proteins that possess these groups.

Paraformaldehyde fixation of a different, but related system, the MHC class II antigen I-A^d on A20 cells, crosslinked these proteins into larger-scale structures that, in fluorescence photobleaching measurements, underwent a substantial reduction in lateral mobility but did not exhibit complete lateral immobilization. They also displayed apparent rotational immobility on the timescale of the time-resolved phosphorescence anisotropy technique used [28]. The lateral diffusion constant decreased monotonically with increasing concentrations of paraformaldehyde yet the fractional recovery in these photobleaching experiments remained constant. This suggests that, though the receptors would have had to have been crosslinked into much larger complexes to exhibit such a decrease in diffusion, they were not completely immobile. Otherwise they would not have diffused at a detectable size-dependent rate and contributed to the fluorescence recovery. Increasing concentrations of paraformaldehyde produced a corresponding increase in the limiting anisotropy of the receptor while not affecting the rotational correlation time on the timescale of these time-resolved phosphorescence anisotropy measurements. The constant correlation time observed is reflective of the portion of unperturbed, non-crosslinked receptors. The fact that the correlation time did not increase on this timescale suggests that, whatever the size of complex formed, it was likely to be many-fold larger than the monomeric protein. The increase in the limiting anisotropy suggests an increased fraction of large crosslinked complexes that are immobile on the μs timescale. Based on these observations, it was concluded that these molecules were crosslinked into aggregates that were large enough to appear immobile on the timescale of time-resolved phosphorescence anisotropy measurements and to exhibit a slower lateral diffusion constant but were nevertheless not large enough to appear laterally immobile on the timescale of the photobleaching measurements. Based on Saffman-Delbrück calculations and the observed decrease in the lateral diffusion constant, the aggregates formed upon treatment

with 0.2% paraformaldehyde may be on the order of 250-fold larger than those not treated with paraformaldehyde. The time-resolved phosphorescence anisotropy measurements implied that the rotational correlation time must have increased at minimum 10-fold and thus put a lower bound on the molecular weight of these complexes of 500,000 kDa. Similarly, it is anticipated that FcεRI expressed on the surface of cells treated with paraformaldehyde would exhibit the slowest rotational correlation times of our five treatments as paraformaldehyde fixation was expected to result not only in a substantial reduction in lateral diffusion but also near complete rotational immobilization of the receptor.

The paraformaldehyde-treated group displayed an average rotational correlation time of 151 ms with a standard error of 42 ms. This value is slightly elevated relative to that obtained for the IgE-only-treated group. Since we expect these receptors to be immobile and to not exhibit any decay in the polarization fluctuation TACF, we do not attribute these findings to crosslinking by paraformaldehyde, but instead once again attribute them to the movement of the lipid rafts to which the receptors are confined. As was the case for the DNP-BSA-treated group, the paraformaldehyde group exhibits a slightly reduced $G_p(0)$ relative to the IgE-only-treated group of 0.00135 ± 0.00091 . This could once again be due to the immobilization of a fraction of receptors by paraformaldehyde.

2.4.5.5. Methyl-β-Cyclodextrin and IgE

Methyl-beta-cyclodextrin (MβCD) is a cyclodextrin compound which directly extracts membrane cholesterol with high specificity. In order to predict the effect such depletion of

membrane cholesterol would have on the rotational diffusion of the FcεRI, it is necessary to consider the effect of cholesterol on membrane fluidity and organization. Cholesterol is an integral component of eukaryotic cell membranes important for membrane structure, organization, function, and dynamics [39, 40]. Cholesterol is an amphipathic molecule with a hydrophilic hydroxyl head in the 3β position oriented towards the exterior of the membrane leaflet and a hydrophobic portion consisting of a planar tetracyclic fused steroid ring and flexible alkyl chain oriented towards the interior of the leaflet so that, as a result, it is aligned with other amphipathic lipids in the bilayer. These structural features are responsible for cholesterol's varied effects on membrane physical properties, such as membrane fluidity [41].

The effects of cholesterol on membrane fluidity are multifaceted. Cholesterol can have seemingly contradictory effects on membrane order depending on the initial state of the membrane. For example, if the membrane is in an intrinsic state of disorder such as is the case in the liquid-disordered (L_d) phase, incorporation of cholesterol tends to confer order to the membrane. A membrane in this phase tends to be enriched in unsaturated phospholipids which, on account of the double bonds present in their fatty acid tails, possess a “kink” in their chains, and thereby tend to reduce lipid packing efficiency. Temperatures significantly above physiological temperature (~37 °C) would also tend to favor such a state. This effect on membrane order can be attributed to the rigid, planar steroid ring of cholesterol which constrains the conformational flexibility of the fatty acid tails of adjacent phospholipids, imparting a more tightly-packed, ordered structure [41]. Conversely, if the membrane is in an inherent state of order as is the case in the solid-ordered phase (S_o), then incorporation of cholesterol will induce disorder in the membrane. A membrane in this phase is typically enriched in saturated

phospholipids and sphingolipids which have a high packing efficiency. Temperatures significantly below physiological temperature (~37 °C) would also tend to favor such a state. In this case, cholesterol disrupts the tight packing of the acyl chains of the saturated phospholipids and sphingolipids [42]. In either case, incorporation of cholesterol converts the membrane to a quasi-crystalline/quasi-liquid state known as the liquid-ordered phase (L_o). It is in this manner cholesterol can both increase membrane fluidity, preventing the membrane from solidifying and maintaining some degree of permeability and plasticity, as well as impart a certain level of order, maintaining the structural integrity of the membrane.

The effects of cholesterol on membrane fluidity are further complicated by the uneven distribution of certain lipids throughout the cell membrane. Tightly-packed saturated phospholipids and sphingolipids have a higher affinity for each other than for unsaturated phospholipids and, as a result, tend to aggregate into small microdomains known as lipid rafts [43], excluding unsaturated phospholipids which remain in the bulk membrane. Because the interaction of cholesterol with these saturated lipids is more favorable [44, 45], the distribution of cholesterol reflects this non-uniform distribution of lipids and therefore it is disproportionately distributed in these lipid rafts. This heterogeneous distribution of cholesterol produces a situation in which immiscible lipid rafts enriched in cholesterol in the L_o phase are suspended in the cholesterol-deficient bulk membrane which exists in the L_d state.

Modulation of membrane cholesterol content can be achieved using various classes of agents including water-soluble sterol-specific carriers that can extract and/or replenish cholesterol from the membrane such as cyclodextrins and liposomes, agents that can sequester

cholesterol in the membrane such as digitonin and nystatin, enzymatic oxidation by cholesterol oxidase, and agents that interfere with cholesterol biosynthesis via inhibition of hydroxymethylglutaryl-coenzyme A (HMG-CoA), which are collectively known as statins [46]. In this study, we have utilized the cyclodextrin known as methyl-beta-cyclodextrin (M β CD). M β CD is an oligomer that consists of seven residues of methylated-glucose. Compared to other cyclodextrins, it exhibits a higher specificity for cholesterol and extracts other lipids to a lesser degree. Previous studies have shown that treatment with increasing concentrations of M β CD generates a progressive reduction in levels of membrane cholesterol while simultaneously having an insignificant effect on the total phospholipid content of the membrane [47]. M β CD binds cholesterol with a stoichiometry of 2:1. Substitution of the polar hydroxyl groups of glucose in M β CD with methyl groups, where typically 10.5-14.7 such substitutions are made per molecule of β -cyclodextrin, imparts a hydrophobic character to the side of the molecule that will interact with cholesterol, while the remaining unsubstituted hydroxyl groups allow it to retain its overall water-soluble character. It is in this way that two molecules of M β CD can “sandwich” cholesterol, selectively extracting cholesterol without partitioning into the membrane. An advantage of cyclodextrins over other lipid carriers such as liposomes [48] or lipoproteins [49] are their small size which enables them to come into close proximity to the membrane and minimize the extent to which the hydrophobic cholesterol molecule experiences the energetically incompatible aqueous environment [50].

Considering the above discussion, it is now possible to consider what effect membrane cholesterol depletion by M β CD may have on the rotational diffusion of the Fc ϵ RI receptor. It has been reported that, unlike other membrane cholesterol-depleting agents, M β CD slows the lateral

diffusion of membrane proteins [51]. However, the effect this agent has on rotational diffusion of membrane proteins, and, in particular, the FcεRI receptor remains uncertain. Given that we would expect IgE-bound FcεRI to be localized in lipid rafts so as to allow for signal transduction and eventual receptor internalization, the effect of such a treatment on the fluidity of the rafts to which these receptors are confined is of interest. Since, as previously discussed, these domains exist in the L_o state, depletion of membrane cholesterol might be expected to cause them to revert to the S_o state. Based on this, it can be hypothesized that cholesterol depletion with MβCD would place the receptor in a more ordered surrounding environment with lowered fluidity, thereby causing the receptor to undergo slower rotational motion. A hypothesis in which treatment with MβCD causes spontaneous desorption of cholesterol from rafts thereby lowering their fluidity is also consistent with the observation from Shvartsman *et al.* (*vide supra*) that MβCD slows lateral diffusion [51]. Thus a scenario can be imagined in which a localized decrease in membrane fluidity in lipid rafts upon MβCD-mediated cholesterol efflux leads to both slowed lateral and rotational diffusion of raft-localized IgE-bound FcεRI.

We observed an average rotational correlation time of 92 ms with a standard error of ±12 ms for the group treated with IgE followed by MβCD. This was the fastest rotational correlation time observed of all groups and was significantly faster than the average rotational correlation time displayed by the group treated with solely IgE. Correspondingly, the histogram of rotational correlation times for the MβCD-treated group shows more QDs in the 30-100 ms bin than for the preceding treatments with >80% of QDs exhibiting rotational correlation times in the 30-300 ms range and relatively fewer QDs exhibiting rotational correlation times >300 ms. This is somewhat surprising given our expectation that depletion of membrane cholesterol with MβCD

would lead to slower rotational diffusion of the receptor. We had based this on a model in which cholesterol at low temperatures imparts a degree of disorder and fluidity to rafts and thus depletion of membrane cholesterol would be predicted to cause rafts to transition to a more ordered, less fluid state. We observed an amplitude and associated standard deviation for the polarization fluctuation TACF of $(1.23 \pm 0.98) \times 10^{-3}$.

2.4.5.6. Cytochalasin D and IgE

Cytochalasin D is a disrupter of actin filament polymerization. It has been demonstrated that cytochalasin D increases FcεRI-mediated degranulation. This suggests that actin microfilaments interact with the receptor in either a direct or indirect manner and down-regulate FcεRI-mediated signal transduction, possibly through a mechanism involving an alteration in the degree of receptor tyrosine phosphorylation taking place [52]. It has also been shown that, upon disruption of actin polymerization with cytochalasin D, crosslinking of FcεRI causes lipid raft components to coalesce on RBL-2H3 mast cells [53].

In fluorescence photobleaching recovery measurements, cytochalasin D has been found to increase lateral diffusion and the observed mobile fraction of hCG-occupied luteinizing hormone (LH) receptor, a G-protein coupled receptor studied extensively in our lab [54]. It was observed that untreated hCG-occupied LH receptor possessed a lateral diffusion constant of $3.3 \times 10^{-11} \text{ cm}^2 \text{ s}^{-1}$ and a mobile fraction of 10.3% and, upon treatment with cytochalasin D, these values underwent a substantial increase to $22.0 \times 10^{-11} \text{ cm}^2 \text{ s}^{-1}$ and ~25-30% respectively. It was also observed that the receptor exhibits a bigger lateral diffusion coefficient and mobile fraction

on regions of the membrane which are physically separated from the underlying cytoskeletal scaffolding known as blebs. These two observations suggest that, when intact, actin filaments tether/anchor the receptor in place and/or constrain its motion/coral the receptor via boundaries dictated by actin fences. Cytochalasin D, as a disrupter of actin filament polymerization, would then be expected to liberate the receptor which would explain its increased lateral mobility. Whether such effects could be generalized to other membrane receptors and thereby extended to the FcεRI remains to be determined, but seems likely.

We report an average rotational correlation time of 111 ms with a standard error of ± 42 ms for the group treated with cytochalasin D. This is a similar yet slightly reduced value than we observed for the IgE-only treated group. In support of this, the histogram of rotational correlation times exhibits a slightly left-skewed distribution when compared to the IgE-only-treated group. We observed an average polarization fluctuation TACF amplitude of $(1.43 \pm 1.29) \times 10^{-3}$ for the cytochalasin D-treated group.

2.4.6. Comparisons of analysis programs

Our new primary analysis program differed from the initial old analysis program in several respects. One of these differences related to how we optimized the values of experimental parameters such as the g-factor and the backgrounds. As we mentioned previously, a major aim of this optimization was to minimize the dependence of the polarization fluctuation TACF on the intensity fluctuation TACF in order to prevent feedthrough of blinking into the polarization fluctuation TACF. In mathematical terms, this meant we wanted a correlation

coefficient between these two quantities as close to zero as possible, which would imply that there was no interdependence between them. In the old method, this meant calculating a conventional Pearson correlation coefficient. In the new method, we employed the Szekeley distance correlation coefficient.

Another difference between these programs was in how we weighted points during the fitting of the polarization fluctuation TACF. In the initial primary analysis program, we used a $s_0^2(1+p_0^2)$ weighting scheme whereas the new program permitted several different weighting modes including 1. a uniform weighting mode; 2. a s^2 weighting mode; and 3. a full weighting mode using individual point values of s and p . For the majority of our analyses, we employed mode 2 - that is, an s^2 weighting mode. Another difference was in the number of parameters used for the fit. Whereas both programs could execute a two-parameter fit with no limiting polarization and a three-parameter fit which included a limiting polarization, in the initial program a separate add-on program was required in order to do a three-parameter fit whereas in the final program this could be specified at the end of the program.

In both the initial and final programs we employed for correlation calculations a fast Fourier transform-based function in Mathematica. This expedited the analysis time so that instead of the time being proportional to N^2 where N is the number of points to be correlated, it is instead proportional to $N \log N$. This shortened what would otherwise have been a prohibitively long analysis time considerably.

2.4.7. Theoretical Expectations for Polarization Fluctuation TACF Amplitudes

In order to understand the significance of the magnitudes of the polarization fluctuation TACF amplitudes, it is necessary to provide some theoretical context. Previous results from our lab report values for the initial anisotropy of solution phase QD605 and QD655 of 0.079 and 0.042 respectively [5]. Grecco *et al.* report a similar value of 0.085 for the initial anisotropy of QD 605 [55]. Theory predicts that these dots should be expected to display initial anisotropies in solution of 0.200 in accordance with the following equation

$$r(t) = \frac{f}{5} e^{-6D_r t} = r_0 e^{-6D_r t} \quad (2.38)$$

The lower values obtained in our experiments could reflect the transition of those QDs. In water, $G_p(0)$ is equal to 0.07. Squaring this gives 0.0049. The real anisotropy is 5/12 of the steady-state anisotropy which gives 0.0020.

2.4.8. Theoretical Estimations for Rotational Correlation Times

The magnitudes of the rotational correlation times obtained using the fluorescence imaging technique described herein merit comment. In order to put these values in context, it is worthwhile to consider previous results regarding this topic. Our lab has previously monitored the rotational dynamics of erythrosin-tagged IgE-FcεRI expressed on 2H3 cells using time-resolved phosphorescence anisotropy [27]. In these experiments, the receptor was observed to exhibit a rotational correlation time of 82 μs which was taken to reflect the hydrodynamic rotation of the receptor. Such short timescales are inaccessible by our imaging approach because the ability of this technique to resolve such information is inherently limited by the frame rate of the camera used. This limitation is addressed in Chapter III of this thesis by our alternative

approach of time-tagged single photon counting. However, the appeal of the imaging approach used here is its ability to access information on a timescale longer than that which is accessible by time-resolved phosphorescence anisotropy methods. The time regime accessible to time-resolved phosphorescence anisotropy methods is intrinsically limited by the phosphorescent lifetime of the probe which, for probes like erythrosin, are typically $<500 \mu\text{s}$. Consequently, this technique can only yield rotational information on timescales below that of the phosphorescence lifetime such that rotational motion occurring on a timescale greater than $\sim 500 \mu\text{s}$ cannot be observed. However, what is observed in such measurements is a limiting anisotropy at $350 \mu\text{s}$ after excitation.

While this residual anisotropy could be interpreted as reflecting a rotationally immobile fraction of receptors, we suggest that it is instead indicative of rotational motions on a longer timescale. We propose that these ms motions observed using our imaging technique here do not correspond to the hydrodynamic rotation of the receptor itself, but instead reflect the rotational dynamics of the larger-scale compartments to which the receptors are confined. It is well known that the cell membrane has a heterogeneous composition and organization, and these motions could reflect the motions of mesoscale heterogeneities distributed throughout the membrane known as lipid rafts that have been discussed previously. These domains may undergo rotational or librational motion and the receptor would, by virtue of being contained within such a region, also be subjected to such ms global membrane motions which are superimposed on its own individual rotations occurring on a μs time regime. Theoretical calculations can give an approximation of how big such a region would need to be in order to produce rotational

motions on such a timescale. Rearrangement of Equation 1.5 and substitution of the diffusion constant with the rotational correlation time via the equation $D_R=1/6\phi$ gives:

$$\rho = \sqrt{\frac{6\phi k_B T}{4\pi\mu h}} \quad (2.39)$$

where ρ is the radius, k_B is Boltzmann's constant, T is the temperature, μ is the membrane viscosity, and h is the membrane thickness. If we use typical values for the membrane thickness and viscosity of 5 nm and 3 poise respectively, a temperature of 37 °C (physiologic temperature), and an rotational correlation time of ~125 ms as was typical in our experiments, this would give a radius of ~413 nm. However, since our experiments were conducted at room temperature (25 °C) without a warm water bath, this value is expected to be a little smaller:

$$\rho = \sqrt{\frac{6 \times 0.125 s \times 1.38 \times 10^{-23} \text{ kg m}^2 \text{ s}^{-2} \text{ K}^{-1} \times 298.15 \text{ K}}{4\pi \times 0.1 \times 3 \text{ kg m}^{-1} \text{ s}^{-1} \times 5 \times 10^{-9} \text{ m}}} \times 1 \times 10^9 \text{ nm} \sim 405 \text{ nm} \quad (2.40)$$

Thus, according to the Saffman-Delbrück equation, to give a rotational correlation time of 125 ms, a body embedded in a membrane would need to possess a radius of around 400 nm. This value lies within the estimated range of plasma membrane compartment sizes.

2.5. Conclusions

As demonstrated in this study, correlation analysis of polarization by fluorescence imaging is a viable method for examining membrane protein rotation, and can elucidate

processes such as molecular aggregation events and reveal information on the local environment of these proteins. We have demonstrated that this technique produces reproducible results and that rotation can be examined with this technique even in the presence of complications such as QD blinking and receptor lateral diffusion. We also conducted experiments with both polarized and non-polarized excitation and concluded that these two types of experiments did not give a statistically significant difference in results. We examined the effect of a variety of cell treatments which we expected to change rotational dynamics of the FcεRI on the rotation of this receptor. We observed rotational correlation times scattered within a range of ~ 10 - 10^3 ms amongst individual copies of the receptor with weighted geometric mean rotational correlation times within the ~ 100 - 150 ms range. As these average correlation times displayed little variability between the different treatments, we attributed them as representing not the rotation of the receptor itself, but rather rotation or libration of the domains to which the receptors are confined, the dimensions of which we have attempted to calculate. This phenomenon, presumably present for all treatments (with the possible exception of MβCD as it alters membrane composition and likely organization) does not preclude the possibility of larger receptor complexes as would be observed upon treatment with DNP-BSA or paraformaldehyde, and such higher-order complexes could also contribute to these averages given they rotate on a sufficiently long timescale.

2.6. References

1. Pinaud, F., et al., *Probing cellular events, on quantum dot at a time*. Nature Methods, 2010. **7**: p. 2275-285.
2. Li, J., K. Kwok, and N. Cheung, *Determination of the dipole geometry of fluorescent nanoparticles by polarized excitation and emission analysis*. Applied Spectroscopy, 2016. **70**(2): p. 302-311.
3. Shapiro, I., *Observation of Single-Molecule Rotational Diffusion at Microsecond Timescales by Polarized Fluorescence Correlation Spectroscopy*, in *Chemistry*. 2009, California Institute of Technology: Pasadena, CA. p. 357.
4. Barisas, B., *Rotation of Single Cell Surface Protein Molecules Studied via Nanoparticle Probes*. National Science Foundation. p. 1-20.
5. Zhang, D., et al., *Rotation of Single Cell Surface Receptors Examined by Quantum Dot Probes*, in *Membrane Organization and Dynamics*, A. Chattopadhyay, Editor. 2017, Springer International Publishing. p. 287-309.
6. Whaley, P., *Qdot conjugates: sensitive, multicolor, stable fluorescence*. 2006, Invitrogen Corporation: Carlsbad California.
7. Cantor, C. and P. Schimmel, *Techniques for the study of biological structure and function*, in *Biophysical chemistry II*, W. Freeman, Editor. 1980: San Francisco. p. 441.
8. Rahman, N., et al., *Rotational dynamics of Fc receptors on individually-selected rat mast cells studied by polarized fluorescence depletion*. Biophysical Journal, 1992. **161**: p. 334-361.

9. Nirmal, M., et al., *Fluorescence intermittency in single cadmium selenide nanocrystals*. Nature, 1996. **383**(6603): p. 802-804.
10. Rosenthal, S., et al., *Biocompatible quantum dots for biological applications*. Chemical Biology, 2011. **18**(1): p. 10-24.
11. Fu, Y., J. Zhang, and J. Lakowicz, *Suppressed blinking in single quantum dots (QDs) immobilized near silver island films (SIFs)*. Chemical Physics Letters, 2007. **447**(1-3): p. 96-100.
12. Shimizu, K., et al., *Blinking statistics in single semiconductor nanocrystal quantum dots*. Phys Rev B, 2001. **63**(20): p. 205-316.
13. Efros, A. and M. Rosen, *Random telegraph signal in the photoluminescence intensity of a single quantum dot*. Phys Rev Lett, 1997. **78**: p. 1110-1113.
14. Frantsuzov, P., S. Volkan-Kacso, and B. Janko, *Model of fluorescence intermittency of single colloidal semiconductor quantum dots using multiple recombination centers*. Phys Rev Lett, 2009. **103**.
15. Voznyy, O. and E. Sargent, *Atomistic model of fluorescence intermittency of colloidal quantum dots*. Phys Rev Lett, 2014. **112**: p. 157401-1-157401-5.
16. Fridman, W., *Fc receptors and immunoglobulin binding factors*. FASEB Journal, 1991. **5**(12): p. 2684-2690.
17. Riske, F., et al., *High Affinity Human IgE Receptor (FcepsilonRI)*. The Journal of Biological Chemistry, 1991. **266**: p. 11245-11251.
18. Kumar, V., et al., *Robbins Basic Pathology*. 9th ed. 2012: Saunders.
19. Daeron, M., *Fc Receptor Biology*. Annual Review of Immunology, 1997. **15**: p. 203-234.

20. Sari, S., *Development of an in vitro model system for studying the interaction of Equus caballus IgE with its high-affinity Fc receptor*, in *Department of Molecular Biology and Biotechnology*. 2011, The University of Sheffield. p. 296.
21. Janeway, C., et al., *Immunobiology: The Immune System in Health and Disease*. 5th ed. 2001: Garland Science.
22. Ochiai, K., et al., *A review on Fc epsilon RI on human epidermal Langerhans cells*. 1994. p. 63-64.
23. Gounni, A., et al., *IgE, mast cells, basophils, and eosinophils*. *Journal of Allergy and Clinical Immunology*, 2006. **117**: p. 450-456.
24. Pawankar, R., *Mast cells as orchestrators of the allergic reaction: the IgE-IgE receptor mast cell network*. *Current Opinion in Allergy and Clinical Immunology*, 2001. **1**(1): p. 3-6.
25. Gilfillan, A. and C. Tkaczyk, *Integrated signaling pathways for mast-cell activation*. *Nature Reviews Immunology*, 2006. **6**: p. 218-230.
26. von Bubnoff, D., et al., *The central role of Fc epsilon RI in allergy*. *Clinical and Experimental Dermatology*, 2003. **28**(2): p. 184-187.
27. Song, J., et al., *Time-resolved phosphorescence anisotropy studies of the mast cell function-associated antigen and its interactions with the Type I Fc epsilon Receptor*. *Biochemistry*, 2002: p. 880-889.
28. Barisas, B., et al., *Dynamics of molecules involved in antigen presentation: effects of fixation*. *Molecular Immunology*, 1999. **36**(11-12): p. 701-708.
29. Zhang, D., *Rotational motion and organization studies of cell membrane proteins*, in *Chemistry*. 2016, Colorado State University: Fort Collins, Colorado. p. 236.

30. I, C., *Qdot biotin conjugates user manual*, I. Corporation, Editor. 2006: Carlsbad California.
31. Barisas, B., et al., *Compartmentalization of the Type I Fc epsilon receptor and MAFA on 2H3 cell membranes*. Biophysical Chemistry, 2006. **126**: p. 209-217.
32. Song, J., et al., *Interactions of the mast cell function-associated antigen with the type I FcEpsilon receptor*. Molecular Immunology, 2001. **38**: p. 1315-1321.
33. Rahman, N., et al., *Rotational dynamics of type I FcEpsilon receptors on individually-selected rat mast cells studied by polarized fluorescence depletion*. Biophysical Society, 1992. **61**: p. 334-346.
34. Molecular Probes, I., *Bovine Serum Albumin Conjugates*, I.D. Technologies, Editor. 2005: Molecular Probes, Inc. p. 1-2.
35. Scientific, T. *DNP-BSA (Albumin from bovine serum (BSA), 2,4-dinitrophenylated)*. [cited 2021 03/14].
36. Carsten, M. and H. Eisen, *The interaction of dinitrobenzene derivatives with bovine serum albumin*. Journal American Chemical Society, 1953. **75**: p. 4451-4456.
37. Walker, J., *Formaldehyde*. Reinhold Publishing Company: New York. p. +701.
38. French, D. and J. Edsall, *The reaction of formaldehyde with amino acids and proteins*, in *Advances in Protein Chemistry*, M. Anson and J. Edsall, Editors. 1945, Academic Press: New York. p. 277-335.
39. Simons, K. and E. Ikonen, *How cells handle cholesterol*. Science, 2000. **290**: p. 1721-1726.
40. Mouritsen, O. and M. Zuckermann, *What's so special about cholesterol?* Lipids, 2004. **39**: p. 1101-1113.

41. Yeagle, P., *Cholesterol and the cell membrane*. Biochim Biophys Acta, 1985. **822**: p. 267-287.
42. Barenholz, Y., *Sphingomyelin and cholesterol: from membrane biophysics and rafts to potential medical applications*. Subcell Biochem, 2004. **37**: p. 167-215.
43. Simons, K. and E. Ikonen, *Functional rafts in cell membranes*. Nature, 1997. **387**: p. 569-572.
44. Ohvo-Rekila, H., et al., *Cholesterol interactions with phospholipids in membranes*. Prog Lipid Res, 2002. **41**: p. 66-97.
45. Holthuis, J., et al., *The organizing potential of sphingolipids in intracellular membrane transport*. Physiol Rev, 2001. **81**: p. 1680-1723.
46. Pucadyil, T. and A. Chattopdhyay, *Role of cholesterol in the function and organization of G-protein coupled receptors*. Progress in Lipid Research, 2006. **45**: p. 295-333.
47. Pucadyil, T. and A. Chattopadhyay, *Cholesterol modulates ligand binding and G-protein coupling to serotonin 1A receptors from bovine hippocampus*. Biochim Biophys Acta, 2004. **1663**: p. 188-200.
48. Hartel, S., H. Diehl, and F. Ojeda, *Methyl-beta-cyclodextrins and liposomes as water-soluble carriers for cholesterol incorporation into membranes and its evaluation by a microenzymatic fluorescence assay and membrane fluidity-sensitive dyes*. Anal Biochem, 1998. **258**: p. 277-284.
49. Kilsdonk, E., et al., *Cellular cholesterol efflux mediated by cyclodextrins*. J Biol Chem, 1995. **270**: p. 17250-17256.
50. Rothblat, G., et al., *Cell cholesterol efflux: integration of old and new observations provides new insights*. J Lipid Res, 1999. **40**: p. 781-796.

51. Shvartsman, D., et al., *Cyclodextrins but not compactin inhibit the lateral diffusion of membrane proteins independent of cholesterol*. *Traffic*, 2006. **7**(7): p. 917-926.
52. Frigeri, L. and J. Apgar, *The role of actin microfilaments in the down-regulation of the degranulation response in RBL-2H3 Mast Cells*. *The Journal of Immunology*, 1999. **162**: p. 2243-2250.
53. Holowka, D., E. Sheets, and B. Baird, *Interactions between Fc(epsilon)RI and lipid raft components are regulated by the actin cytoskeleton*. *J. Cell Sci.*, 2000. **113**: p. 1009-1019.
54. Roess, D., G. Niswender, and B. Barisas, *Cytochalasins and colchicine increase the lateral mobility of human chorionic gonadotropin-occupied luteinizing hormone receptors on ovine luteal cells*. *Endocrinology*, 1988. **122**(1): p. 261-269.
55. Grecco, H., et al., *Ensemble and single particle photophysical properties (two-photon excitation, anisotropy, FRET, lifetime, spectral conversion) of commercial quantum dots in solution and in live cells*. *Microsc Res Tech.*, 2004. **65**(4-5): p. 169-179.

Chapter III: Examination of rotation of quantum dot-labeled type I Fc ϵ receptor by time-tagged single photon counting

3.1. Overview

Correlation analysis of polarized optical signals emanating from quantum dot (QD) probes attached to membrane receptors can reveal information on the rotation of these receptors. In this study, we have used a time-tagged single photon counting approach to examine receptor rotation on μ s timescales. In this method, a focused laser excites individual receptor-bound QD displayed on the cell membrane and the resultant fluorescence emitted by the QD polarized parallel and perpendicular with respect to the polarization of the excitation source is collected by separate confocal avalanche photodiode (APD) detectors and thereby sorted into two channels depending on the polarization of the signal. A Becker and Hickl DPC 230 digital time-tagged single photon counter stores information on both arrival time of a detected photon (with a resolution of 165 ps) and the channel in which it arrived. The auto- and cross-correlations of the vertically- and horizontally- polarized signals are determined directly from the arrival times without binning and corrected using a number of experimental parameters such as the g-factor. Careful selection of these parameters was necessary in order to minimize the dependence of the polarization fluctuation time-autocorrelation function (TACF) on the intensity TACF and prevent feed-through of blinking into the polarization TACF. The minimum rotational correlation time accessible to this technique is determined by the maximum detected photon count rate that can be attained. Given typical count rates, correlation times as low as 20 μ s are attainable which should, in theory, enable observation of the hydrodynamic rotation of the receptor. We obtained

correlation times for several QDs that could correspond to the hydrodynamic rotation of the receptor for the IgE-treated group and slightly hindered rotation for the DNP-BSA and paraformaldehyde-treated groups. However, we observed a prevalence of QDs that either displayed a combination of very large correlation times and very small TACF amplitudes or vice versa. Though these may be legitimate indicators of immobility for these QDs, given the ubiquity of QDs exhibiting such behavior throughout all treatment groups, we have considered the possibility that our measurements may have been affected by a systematic error. We speculate on possible causes for such an error, both experimental and in our method of data analysis, and propose possible solutions for removing this complication.

3.2. Introduction

3.2.1. Time-tagged single photon counting approach to measure QD-labeled receptor rotation

In Chapter II, we examined the slow, hindered rotation of the Type I Fc ϵ RI (Fc ϵ RI) on 2H3 rat basophilic leukemia (2H3-RBL) cells using polarized fluorescence imaging of the commercial Quantum Dot 655 (QD655). While our imaging approach is uniquely appealing in that it can potentially reveal rotational phenomena occurring on a previously unexamined ms timescale, the maximum frame rate of available cameras imposes an inherent limitation on the timescale of rotation that can be examined. Though this method is suitable for measuring hindered rotation of this receptor which we attributed to complexation events or localization to lipid microdomains, it is unable to examine rotation on a time-scale faster than several camera frame times, which, for the fastest low-light cameras, is typically on the order of one to several

ms. In our application of this method using an ANDOR EMCCD camera, frames are separated by a minimum of 10 ms and thus it lacks the requisite resolution to reveal phenomena occurring more rapidly than this. Therefore, while this technique may be suitable to examine rotational motion associated with the aforementioned phenomena (*e.g.* rotation of domains in which the receptor is confined or incorporation of the receptor into large aggregates), phenomena that occur on a shorter timescale than this cannot be measured by this method. Importantly, the hydrodynamic rotation of the ligand-bound receptor, which has been measured by time-resolved phosphorescence anisotropy to be 28 μs at 37 °C [1], 40 μs at 25 °C [2], and 82 μs at 4 °C [3] cannot be observed with this technique as these values are far below the minimum threshold accessible to imaging techniques.

To address this limitation, we have implemented an alternative time-tagged single-photon counting approach. These time-tagged single photon counting measurements enable the examination of rotation on a timescale inaccessible to our imaging approach. For this reason, we have also employed an alternative time-tagged single photon counting approach. With this method it is possible to examine rotational motion occurring on a shorter timescale than could be observed with our imaging approach. In theory, given sufficiently high count rates, this technique can be used to probe sub-nanosecond timescales. The clock on our DPC-230 time-tagged single photon counter yields measurements with a resolution of 0.164 ns (164 ps) per point. In practice, given typical maximum photon count rates 50,000 counts/s, this method can probe rotational correlation times $\geq 20 \mu\text{s}$ and therefore has the potential to allow us to observe the hydrodynamic rotation of the receptor.

Unlike our imaging method, the time-tagged single photon counting method is, in theory, capable of detecting the hydrodynamic rotation of the ligand-bound receptor. Because this technique has a resolution of 165 ps, in theory, it is possible to observe rotational correlation times under 1 ns. However, in practice the lower bound on the rotational correlation times we expect to be able to observe is dictated by the maximum number of photons that can be collected over a period of time. Rotation can only be satisfactorily observed if the product of the rotational correlation time with this photon count rate is greater than or equal to one (*i.e.* there is at least one count per correlation time). Given, from a single QD, a maximum signal rate of $\sim 5 \times 10^4$ counts per second, the lower bound on the rotational correlation times we can expect to observe with this technique is $\sim 20 \mu\text{s}$.

Quantum Dot 605 has a lifetime determined by our IBH 5000U lifetime fluorometer to be 5.5 ns [4]. Considering this value, in order to get at least one count per lifetime, we would need $\sim 1.8 \times 10^8$ counts/s which is orders of magnitude above the maximum rate of photons that can be emitted by, much less detected from, an individual QD. This imposes an absolute lower limit on correlation times which might be calculated from such fluorescence signals.

3.2.2. Fluorescence Correlation for Asynchronous Time-Tagged Single-Photon Counting Data

An obstacle that arises when performing single molecule spectroscopy is low photon count rates. This presents a problem in conventional experiments involving synchronous acquisition in which photons are binned into consecutive intervals of equal width. A consequence of the sparse photon streams associated with single molecule methods when

conducting experiments with a synchronous mode of acquisition are empty time bins that collect no photons. This is particularly problematic at high temporal resolution and with the low light intensities which are characteristic of single molecule experiments. In this situation, the small bin widths combined with the rarity of photon detection events means that most bins will contain no information, yet occupy large amounts of memory space. As an example, consider that a bin width of 10 ns would require a count rate of 10^8 photons/s, yet typical count rates tend to be much lower at $\sim 10^3$ - 10^5 photons/s. Hence most bins are empty in this scenario. If we consider a count rate as high as 10^5 photons/s, an experiment with a 30 s measurement time would give an average of 3×10^6 total photons. Given the 10 ns temporal resolution, there would be 3×10^9 bins in total. Assuming each bin requires 4 bits of space, recording all this data would occupy 1.50 GB of space. This is both excessive and unnecessary, as based on the total number of photons collected over the full duration of the experiment, in only 0.1% of these will a photon be recorded whereas the rest of these will be devoid of any valuable information. Given the effective bin width for the raw data in our TTSPC experiments would be the minimal resolution or clock time of 0.164 ns, this excessive overhead of information would be even greater.

An alternative method of data acquisition that addresses the above concern is an asynchronous data acquisition mode known as the time-tagged time-resolved (TTTR) mode [5]. In this mode, unlike with a synchronous mode of data acquisition, there is no sorting of photons based on the time interval in which they arrived and data is only recorded when a photon is detected. A time-tagged single photon counter records two numbers-the arrival time of the photon and the channel in which it was detected (*i.e.* 1 or 2 corresponding to the APDs detecting vertically- and horizontally-polarized light respectively). The obvious advantage of this TTTR

data acquisition mode when compared to the synchronous data acquisition mode is that in this mode the amount of data is directly proportional to the photon count rate. Thus this saves a substantial amount of memory and makes this a suitable method for data acquisition where synchronous acquisition would prove impractical.

3.2.3. Goals of study

In this study, we first aimed to establish our TTSPC method as a viable way of measuring the rotation of cell membrane receptors on a microsecond timescale. Examination of the rotation of the receptor on this shorter timescale would provide complementary information on faster receptor motion and give a more complete picture of receptor rotational dynamics than our imaging approach (the subject of the previous chapter) can alone. Amongst other goals, we hoped this technique would enable us to observe the hydrodynamic rotation of QD-labeled, IgE-bound FcεRI. Additionally, we sought to examine the effect of a number of cell treatments which we anticipated would perturb receptor rotation to a considerable degree. These treatments consisted of IgE plus an additional agent such as DNP-BSA, paraformaldehyde, MβCD, and cytochalasin D. We also experimented with two different types of QD to label the receptor, QD605 and QD655, the goal being to compare the effect of their differing optical properties and aspect ratios on the results obtained.

3.3. Materials and Methods

3.3.1. “Wet” chemistry procedures

Methods for growing RBL-2H3 cells, sensitization of FcεRI with DNP-specific A2-IgE, and labeling with QD655 and QD605 are described in detail in chapter II of this dissertation. See section 2.3.1 above for a full description of these procedures.

3.3.2. Photon-counting data collection and analysis

The experimental apparatus used to conduct time-tagged single photon counting experiments included an Olympus IX-71 microscope equipped with a Becker & Hickel DPC-230 time-tagged single photon counter and an ALV-7004 digital hardware correlator. A schematic of this experimental setup is shown in Figure 3.1. A 488 nm blue ModuLaser StellarPro multiline argon ion laser was used to excite the sample. The laser traveled along a KineFLEX™-P-3-S-488-640-0.7-FCP-P2 fiber optic cable and, upon exiting the cable was re-collimated by a collimating lens. The microscope and laser excitation path are shown in Figure 3.2.

A 0.5X Galilean telescope reduced the beam radius by half. The spot size/detection volume radius on the sample is inversely related to the beam radius by the equation $r = \lambda f / \pi R$ where λ is the emission wavelength, f is the focal length of the objective, and R is the laser beam radius. Thus this 0.5x reduction in laser beam produced a 2x expansion of the beam radius on the sample. The laser beam then passed through an adjustable iris. It was then reflected by a mirror and a gimbal mount positioned at the re-created back focal plane of the objective. This system of mirror and gimbal mount could be adjusted (both the position and pitch and yaw of either mirror) so that the laser was centered on the objective and did not come through at an angle.

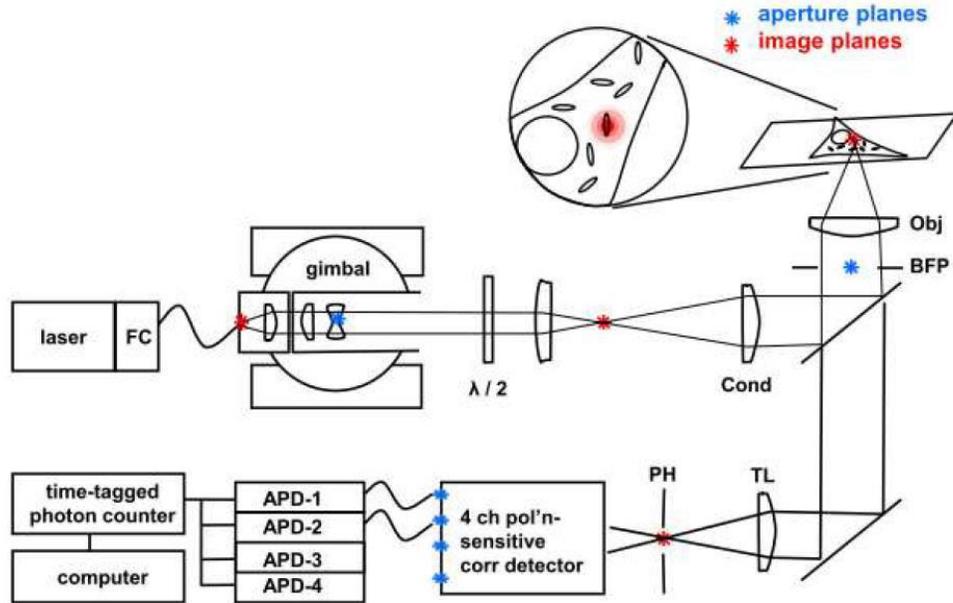


Figure 3.1: Schematic diagram of microscope used for TTSPC measurements. A system consisting of a primary mirror and a secondary gimbal-mounted mirror were used to direct the laser into the microscope. The locations of alternating aperture and image planes are denoted by blue and red asterisks, respectively. A dichroic mirror reflects incident laser light onto the sample and transmits fluorescence emitted from the sample onto the detection path. The fluorescence travels through a pinhole which filters out-of-focus light originating from front or behind the back focal plane of the objective. A custom-built detector box separates this light by polarization and directs it to separate avalanche photodiode (APD) detectors.



Figure 3.2: Olympus IX-71 microscope used in TTSPC measurements and excitation path for argon 488 nm laser. After leaving the laser housing on the left side of the diagram, the laser travels through a fiber optic cable, a collimator, a mounted telescope, and an adjustable iris. It is then reflected by a mirror onto a second mirror on a gimbal mount. The laser beam then enters the port on the left side of the microscope and is directed onto the sample positioned on the adjustable stage. Fluorescence exits the port on the right side of the microscope travels through a pinhole and our custom-built detector box (not pictured).

For these measurements, a mechanism was needed to separate vertically- and horizontally-polarized fluorescence emissions so that they could be individually recorded by our detectors. To accomplish this, we fabricated our own unique custom detector system. Fluorescence polarized in vertical and horizontal orientations was recorded in 2 channels by this detector system which is capable of recording up to 4 channels. If required, these extra 2 detectors could independently record fluorescence of the same polarization examined by one of the aforementioned detectors for use in subsequent pseudo-autocorrelation calculations. For these experiments; however, only 2 detectors were used.

Fluorescence exiting the side port of the microscope entered a 300 μm pinhole positioned at an image plane which filtered out out-of-focus light originating from front or behind the back focal plane of the objective. This fluorescence then entered our custom detector box (see Figures 3.3 and 3.4). After a telescope, a polarizer is positioned in the optical path which separates incident light by polarization. Vertically-polarized light is reflected by the polarizer and travels along a new optical path through a telescope and towards a collimating lens connected by a fiber optic to an APD detector, while horizontally-polarized light is transmitted through the polarizer orthogonally to this optical path and reflected 90° by a mirror onto a third parallel optical path, through another telescope, and onto a second collimating lens connected by a fiber optic to another APD detector. For these experiments, we have only used these two detectors to separately monitor vertically- and horizontally- polarized fluorescence.

It is worth noting that this detection system is, in theory, capable of monitoring up to four channels using four APD detectors. This is accomplished by placing a beamsplitter before

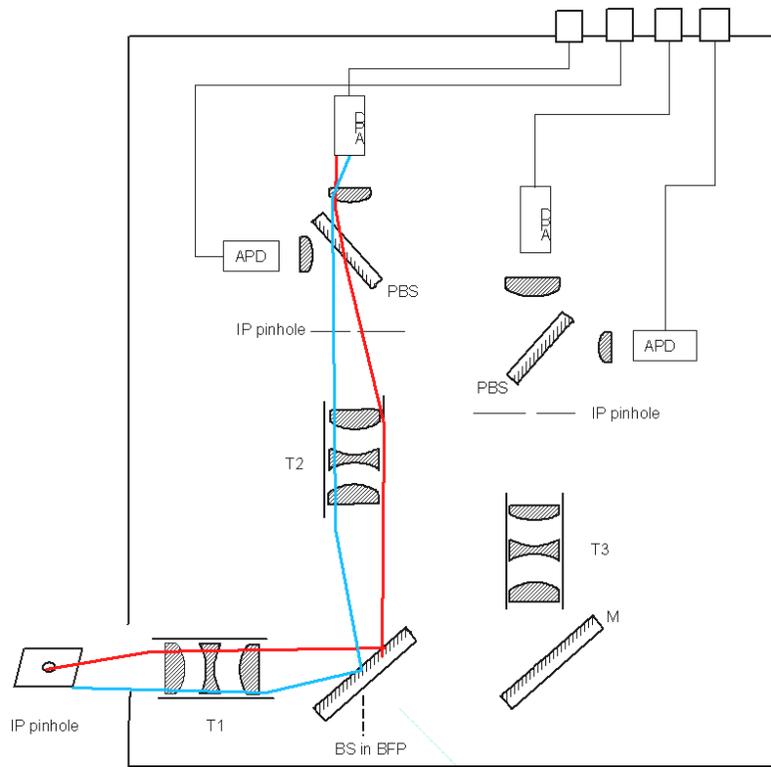


Figure 3.3: Schematic of custom-built detector box. A 50 μm pinhole positioned at an image plane filters out-of-focus fluorescence originating from front or behind the back focal plane of the objective. Fluorescence originating from the sample enters a primary telephoto lens system (T1) consisting of an array of two convex and one concave lenses. A beamsplitter (BS) is placed behind this which can transmit and reflect 50% of incoming light, or, in the case of cross-correlation measurements, a wavelength-partitioning filter can be used. The light then travels through a secondary telephoto lens system (T2) consisting of another array of three lenses. A polarizer/polarizing beamsplitter (PBS) separates vertically- and horizontally-polarized light and directs it onto separate avalanche photodiode (APD) detectors. Fiber optic cables relay the signals to a Becker and Hickl DPC 230 digital photon counter.

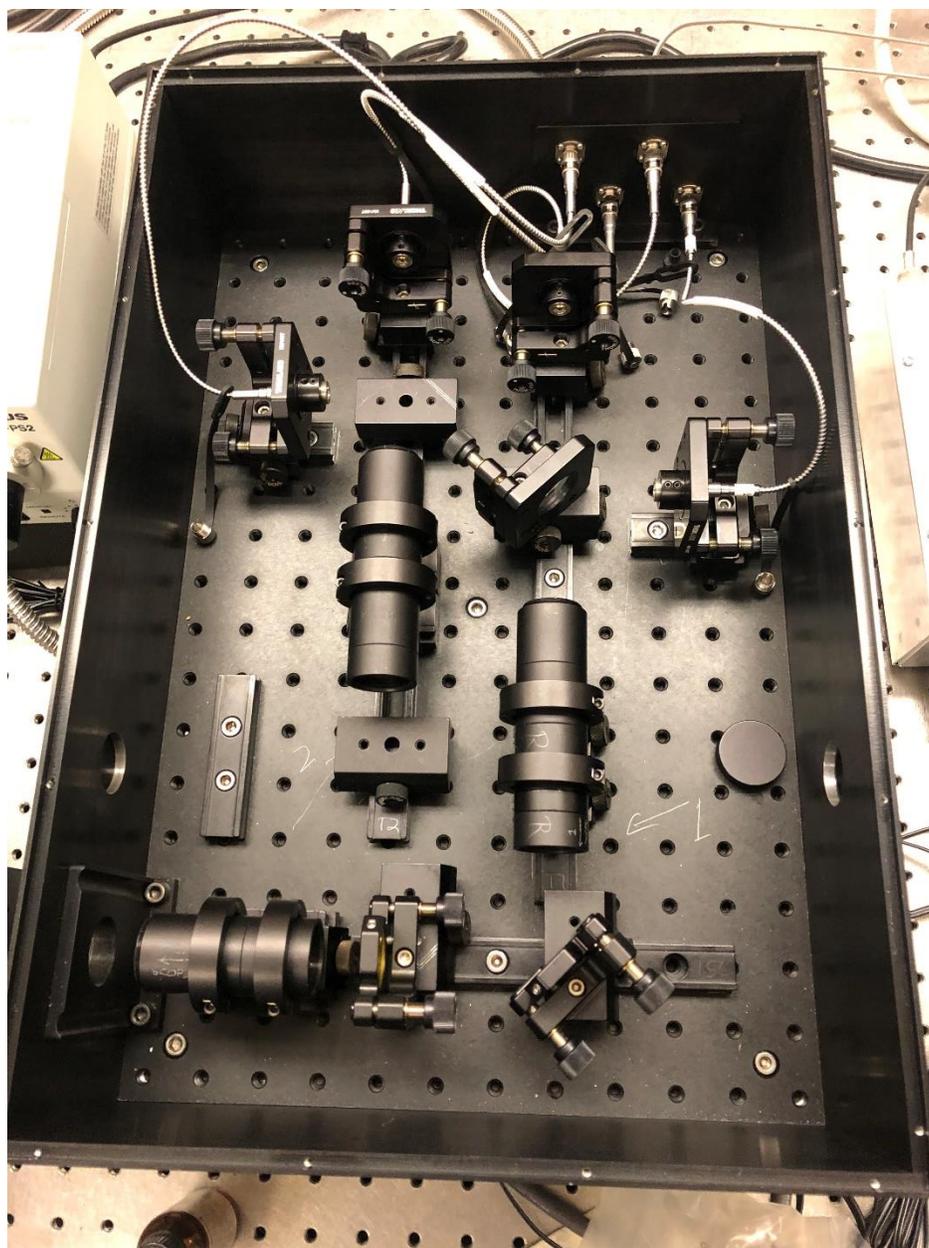


Figure 3.4: Picture of custom-built detector box for TTSPC measurements. The design of this setup is explained in the text and the accompanying caption for the schematic representation in Figure 3.3.

each of the aforementioned detectors. The beamsplitter transmits half of the incident light intensity onto this detector and an additional detector is positioned to collect the other half of the light reflected by the beamsplitter. This is done in a likewise fashion for the other detector. This enables the collection of the vertically-polarized light by two separate detectors. Likewise, horizontally-polarized light can be collected by the other two detectors. In this manner, it is possible to autocorrelate the same signal (at half intensity) collected by two different detectors. This practice, termed “pseudo-autocorrelation”, circumvents problems such as detector-afterpulsing typically associated with conventional autocorrelation of signals recorded by the same detector. This is especially important as it affects early points in the correlation curve and can therefore interfere with the ability to accurately determine amplitudes. However, a drawback to using the additional detectors for pseudo-autocorrelation measurements is that the signal intensity is effectively split in half.

A Becker and Hickl DPC 230 digital photon counter records the arrival time and channel (either 1 or 2) of every photon and has a minimal time resolution of 0.17 ns. The information corresponding to each photon takes up 4 bytes of space in each file.

SPC files generated by the DPC-230 time-tagged single photon counter containing raw data for each QD were processed with a program written in Power Basic to generate CSV files which consisted of tabulated data with a column recording the arrival time in DPC-230 clocks (“time-tag”) of each photon and another column indicating whether the photon was sorted into channel 1 or 2 (*i.e.* whether the photon arrived at the APD detector which detects parallel-polarized or the detector for perpendicularly-polarized signals).

A program written in Mathematica implemented a corrected correlation algorithm based on a principle introduced by Enderlein and colleagues [6]. This program is included in its entirety in the Appendix as Program Code 5: “PC Corr v.80”. This algorithm does not attempt to bin all photons into a set of equal-width bins but rather only processes each photon individually. Consider a 30 s experiment with a photon flux of 5×10^4 photons/s. A file of 1.8×10^{11} bins uniformly-spaced bins, and possibly 4x this number of bytes, would be needed for these data. Since each bin would require mathematical processing, analysis of such a file would be effectively impossible. By contrast, this example would involve 1.5×10^6 photons so that the arithmetic required to process the individual photons but ignoring empty bins would be about 10^5 -fold less than that needed to process each bin.

Our algorithm was used to correlate the time-tagged photon data from the CSV files described previously. A total of nine auto- and cross-correlation functions were calculated as described subsequently. Four of these correlation functions cover the possible combinations of the vertical and horizontal intensity functions including the auto-correlation of the vertical intensity with itself G_{vv} , the cross-correlation of the vertical intensity with the horizontal intensity G_{vh} , the cross-correlation of the horizontal intensity with the vertical intensity G_{hv} , and the auto-correlation of the horizontal intensity with itself G_{hh} . The other five correlations give the averages of v_1 , v_2 , h_1 , and h_2 where v_1 and h_1 are early points in the correlation and v_2 and h_2 are the second points some time τ later. These are equivalent to correlating intensities with an array of ones: G_{v1} , G_{1v} , G_{1h} , G_{h1} and G_{11} , that is, the number of points n . When these correlations are performed using our sparse matrix method, it is much faster than simply summing the huge

sparse arrays. For example, a data file from one of these experiments reflects $\sim 3 \times 10^{11}$ detector clocks and contains entries for 3 million photons in a file ~ 5 -30 mB in length.

All correlation functions were calculated at pseudo-logarithmically-spaced lag times. These lag times were multiples of the minimal clock time of our photon counter which was 0.17 ns. Data were sorted into bins with widths 2^m that are multiples of this fundamental clock time. So for m values of 0, 1, 2, and 3, the bin widths would be 1, 2, 4, and 8 clocks or 0.17 ns, 0.34 ns, 0.68 ns, and 1.36 ns. However, because these timescales are far too short to monitor rotation, we pre-bin raw data by a factor of 2^{10} . With this first bin consisting of 1024 clock times, this gave an initial bin width of $\sim 0.17 \mu\text{s}$. Our algorithm produced lag times spaced with this minimal resolution of $\sim 0.17 \mu\text{s}$. The lag times are spaced this far apart for the first eight lag time points at which point the spacing doubled to $\sim 0.34 \mu\text{s}$ which continued for eight time points at which point the value quadrupled to $\sim 0.68 \mu\text{s}$ between points and so on.

As was the case for the imaging data, several parameters, both instrumental and arising from the sample, were employed to correct these raw correlation functions to obtain the true correlations. These parameters include the instrumental parameter g which, as described in Chapter II, is used to correct for differential detection efficiencies in the two orientations as well as correction constants a and b which account for cell background intensities in the vertical and horizontal orientations respectively. These correction factors contain the same information as the g , b , and c used in processing the imaging data, but are, for historical reasons, written differently for the photon-counting analogues. To correct the observed vertical and horizontal intensities v_o

and h_o to obtain the corrected intensities v_c and h_c , the following equations were used. It proved convenient to replace the quantities $1-g$ and $1+g$ with simpler symbols g_v and g_h respectively.

$$\begin{aligned} v_c &= (1-g)(v_o - a) \\ &= g_v (v_o - a) \end{aligned} \quad (3.1)$$

$$\begin{aligned} h_c &= (1+g)(h_o - b) \\ &= g_h (h_o - b) \end{aligned} \quad (3.2)$$

This correction procedure allows us to calculate the corrected TAC of polarization fluctuations in terms of the observable quantities. For example, obtaining the cross-correlation at time τ of the corrected vertical intensity v_{1c} at time t and the corrected horizontal intensity h_{2c} at some time later $t+\tau$ in terms of the uncorrected observed intensities at these times v_{1o} and h_{2o} is then straightforward:

$$\begin{aligned} G_{vh} &= \sum v_{1c} h_{2c} \\ &= \sum g_v g_h (v_{1o} - a)(h_{2o} - b) \\ &= g_v g_h \sum v_{1o} h_{2o} - b g_v g_h \sum v_{1o} \\ &\quad - a g_v g_h \sum v_{ho} + g_v g_h ab \sum 1 \end{aligned} \quad (3.3)$$

Equations for $v_{1c}v_{2c}$, $h_{1c}v_{2c}$, and $h_{1c}h_{2c}$ were derived in an analogous manner using the equations for v_c and h_c .

Likewise, the sums of the individual signals $\langle v_{1o} \rangle$, *etc.* are effectively the correlation of the signal in question with an array of ones and can be calculated during the correlation process:

$$\begin{aligned} G_{v1} &= \sum (v_{1o} \cdot 1) \\ &= \sum v_{1o} \\ &= g_v \sum v_{1o} - a \sum 1 \end{aligned} \quad (3.4)$$

Obtaining the value of the corrected autocorrelations $G_{vv(c)}$, $G_{vh(c)}$, $G_{hv(c)}$, $G_{hh(c)}$, G_{v1} , G_{1v} , G_{h1} , G_{1h} , and G_{11} for a given value of τ involves evaluating the product of the corrected intensities for every combination of time points separated by a given lag time τ , summing these products, and repeating this for every τ in the correlation curve. We note that we will henceforth use the notations for two points and the respective correlation (e.g. $v_{1c}h_{2c}$ and G_{vh}) interchangeably-where a formula specifies two points, it can be assumed that the respective correlation function that we desire is simply obtained in this manner.

With the four corrected correlations, it was possible to calculate a corrected polarization time autocorrelation function (TACF). As introduced in chapter II, the polarization is defined as:

$$p(t) = \frac{I_{\parallel}(t) - I_{\perp}(t)}{I_{\parallel}(t) + I_{\perp}(t)} \quad (3.5)$$

Where the numerator is simply recognized as the difference in the intensities in the orientations parallel and perpendicular to the polarization of the exciting light and the denominator is the sum of these intensities in these parallel and orthogonal or orthogonal relative to the exciting light. Given this definition, the product of the corrected polarizations p_{1c} and p_{2c} could be calculated as:

$$p_{1c}p_{2c} = \frac{d_{1c}d_{2c}}{s_{1c}s_{2c}} \quad (3.6)$$

Where $d_{1c}d_{2c}$ is the autocorrelation of the corrected difference function and $s_{1c}s_{2c}$ is the autocorrelation of the sum function. The polarization TACF G_{pp} was then calculated as the

quotient of G_{dd}/G_{ss} . To obtain this, it was first necessary to determine the autocorrelation of the difference function G_{dd} and the autocorrelation of the sum function G_{ss} . For G_{dd} , consider the product of the corrected differences d_{1c} and d_{2c} at two different time points for a pair of corrected vertically-polarized v_{1c} and v_{2c} and horizontally-polarized intensities h_{1c} and h_{2c} at those respective points:

$$d_{1c}d_{2c} = (v_{1c} - h_{1c})(v_{2c} - h_{2c}) \quad (3.7)$$

Substituting the observed intensities from equations 3.1 and 3.2 into equation 3.8 for $d_{1c}d_{2c}$ gives the corrected difference function in terms of experimental observables:

$$\begin{aligned} d_{1c}d_{2c} = & g_v^2[v_{1o}v_{2o} - a(v_{1o} + v_{2o}) + a^2] \\ & - g_v g_h (v_{1o}h_{2o} - bv_{1o} - ah_{2o} + ab) \\ & - g_v g_h (h_{1o}v_{2o} - ah_{1o} - bv_{2o} + ab) \\ & + g_h^2[h_{1o}h_{2o} - b(h_{1o} + h_{2o}) + b^2] \end{aligned} \quad (3.8)$$

Consolidating like terms gives:

$$d_{1c}d_{2c} = C_1v_{1o}v_{2o} + C_2v_{1o}h_{2o} + C_3h_{1o}v_{2o} + C_4h_{1o}h_{2o} + C_5v_{1o} + C_6v_{2o} + C_7h_{1o} + C_8h_{2o} + C_9n \quad (3.9)$$

Where the coefficients for each term are:

$$\begin{aligned}
C_1 &= g_v^2 \\
C_2 &= -g_v g_h \\
C_3 &= -g_v g_h \\
C_4 &= g_h^2 \\
C_5 &= -g_v (ag_v - bg_h) \\
C_6 &= -g_v [ag_v + bg_h] \\
C_7 &= g_h [ag_v - bg_h] \\
C_8 &= g_h [ag_v - bg_h] \\
C_9 &= [ag_v - bg_h]^2
\end{aligned} \tag{3.10}$$

The respective corrected difference correlation function is then:

$$G_{dd(c)} = C_1 G_{vv(o)} + C_2 G_{vh(o)} + C_3 G_{hv(o)} + C_4 G_{hh(o)} + C_5 G_{v1} + C_6 G_{1v} + C_7 G_{h1} + C_8 G_{1h} + C_9 G_{11} \tag{3.11}$$

Obtaining this corrected autocorrelation $G_{dd(c)}$ involves evaluating the observed correlation functions in each term (e.g. $G_{vv(o)}$) for every combination of time points separated by a given lag time τ , summing these products and repeating this for every τ in the correlation curve to obtain the respective correlation and summing these first four terms. If we were to simplify by assuming 0-based signal arrays in Mathematica, $G_{dd(c)}$ could be expressed in terms of these first four basic correlations $G_{vv(o)}$, $G_{vh(o)}$, $G_{hv(o)}$, and $G_{hh(o)}$. However, five other combinations (corresponding to the last five terms of equations 3.9 and 3.11) were correlated involving 1-based signal arrays in Mathematica: the correlation of v_{1c} with an array of ones which represents the average of the corrected vertical intensities at time t and will henceforth denoted as $\langle v_{1c} \rangle$, the correlation of v_{2c} with an array of ones which represents the time-average of the corrected vertical intensities at time $t+\tau$ henceforth denoted as $\langle v_{2c} \rangle$, the correlation of h_{1c} with an array of ones which represents the time-average of the corrected horizontal intensities at time $t+\tau$ henceforth denoted as $\langle h_{1c} \rangle$, the correlation of h_{2c} with an array of ones which represents the time-average of the

corrected horizontal intensities at time $t+\tau$ henceforth denoted as $\langle h_{2c} \rangle$, and the correlation of these arrays of ones with each other which simply represents the number of points in the correlation n .

Similarly, for G_{ss} , consider the product of the sums:

$$s_{1c}s_{2c} = (v_{1c} + zh_{1c})(v_{2c} + zh_{2c}) \quad (3.12)$$

This is treated in an analogous manner as $d_{1c}d_{2c}$. Expanding equation 3.10 and substituting the expression for v_c and h_c from equations 3.1 and 3.2, $s_{1c}s_{2c}$ would then be expressed in terms of these more fundamental observed correlations:

$$\begin{aligned} s_{1c}s_{2c} = & g_v^2[v_{1o}v_{2o} - a(v_{1o} + v_{2o}) + a^2] \\ & + zg_v g_h(v_{1o}h_{2o} - bv_{1o} - ah_{2o} + ab) \\ & + zg_v g_h(h_{1o}v_{2o} - ah_{1o} - bv_{2o} + ab) \\ & + z^2 g_h^2[h_{1o}h_{2o} - b(h_{1o} + h_{2o}) + b^2] \end{aligned} \quad (3.13)$$

Consolidating like terms gives:

$$s_{1c}s_{2c} = C_1 v_{1o} v_{2o} + C_2 v_{1o} h_{2o} + C_3 h_{1o} v_{2o} + C_4 h_{1o} h_{2o} + C_5 v_{1o} + C_6 v_{2o} + C_7 h_{1o} + C_8 h_{2o} + C_9 n \quad (3.14)$$

where the prime is intended to differentiate these new constant coefficients from those obtained earlier in our derivation of the expression for $d_{1c}d_{2c}$. The expressions for these new constants are as follows:

$$\begin{aligned}
C_{1'} &= g_v^2 \\
C_{2'} &= zg_v g_h \\
C_{3'} &= zg_v g_h \\
C_{4'} &= g_h^2 \\
C_{5'} &= -g_v(ag_v + zbg_h) \\
C_{6'} &= -g_v[ag_v + zbg_h] \\
C_{7'} &= -zg_h(ag_v + zbg_h) \\
C_{8'} &= -zg_h(ag_v + zbg_h) \\
C_{9'} &= (ag_v + zbg_h)^2 n
\end{aligned} \tag{3.15}$$

The respective corrected intensity sum correlation function is then:

$$G_{ss(c)} = C_{1'}G_{vv(o)} + C_{2'}G_{vh(o)} + C_{3'}G_{hv(o)} + C_{4'}G_{hh(o)} + C_{5'}G_{v1} + C_{6'}G_{1v} + C_{7'}G_{h1} + C_{8'}G_{1h} + C_{9'}G_{11} \tag{3.16}$$

While the expression we have derived above can be used in the calculation of either the anisotropy ($z=2$) or the polarization ($z=1$) TACF, we have opted to use the polarization TACF in these studies. Since we have continued to use polarization as we did in the imaging project, z is equal to 1 and can be omitted in the above equation. The product of the corrected polarizations p_{1c} and p_{2c} and respective autocorrelation G_{pp} was then calculated as the quotient of the respective intensity difference function divided by the intensity sum function as introduced previously in this section.

Because we required the TACF of the polarization *fluctuations*, this expression had to be modified slightly to give the autocorrelation of the fluctuations of the polarization from the mean polarization. This polarization fluctuation TACF is defined as follows:

$$G(p, \tau) = \frac{1}{m-i+1} \sum_{i=1}^{m-\tau} (p_1(i) - \overline{p_1})(p_2(i+\tau) - \overline{p_2}) \quad (3.17)$$

For each τ in the correlation trace, the product of the fluctuations is evaluated for all m and these are summed to give the value of $G(p, \tau)$. The upper bound is adjusted from m to $m-\tau$ because no complementary $p_2(i+\tau)$ exists for a $p_1(i)$ when $i=m$.

This expression can also be written in an alternative notation with $\Delta p_1 = p_1(n) - \langle p_1 \rangle$ and $\Delta p_2 = p_2(n) - \langle p_2 \rangle$ for brevity:

$$G(p, \tau) = \frac{1}{m-i+1} \sum_{i=1}^{m-\tau} (\Delta p_1(i))(\Delta p_2(i+\tau)) \quad (3.18)$$

The constant product of the averages can be taken out of the summation as follows:

$$G(p, \tau) = \left(\frac{1}{m-i+1} \sum_{i=1}^{m-\tau} p_1(i)p_2(i+\tau) \right) - \overline{p_1 p_2} \quad (3.19)$$

Four of the nine correlation functions derived in the preceding section were used to derive the polarization autocorrelation. The polarization autocorrelation G_{pp} was used to give the polarization fluctuation TACF $G_{pp} - \langle p_{1c} \rangle \langle p_{2c} \rangle$ where the average polarizations $\langle p_{1c} \rangle$ and $\langle p_{2c} \rangle$ were given by the following equations:

$$\overline{p_{1c}} = \frac{\overline{v_{1c}} - \overline{h_{1c}}}{\overline{v_{1c}} + \overline{h_{1c}}} \quad (3.20)$$

$$\overline{p_{2c}} = \frac{\overline{v_{2c}} - \overline{h_{2c}}}{\overline{v_{2c}} + \overline{h_{2c}}} \quad (3.21)$$

And the corresponding equations in terms of the correlations:

$$\overline{p_{1c}} = \frac{G_{v1} - G_{h1}}{G_{v1} + G_{h1}} \quad (3.22)$$

$$\overline{p_{2c}} = \frac{G_{1v} - G_{1h}}{G_{1v} + G_{1h}} \quad (3.23)$$

The product of $\langle p_{1c} \rangle$ and $\langle p_{2c} \rangle$ was then subtracted from the polarization TACF G_{pp} to give the polarization fluctuation TACF.

We calculated a residual χ for every point based upon the following equation:

$$\chi = G_{dd} - G_{ss} \times \overline{p_{1c} p_{2c}} \quad (3.24)$$

Optimization of the polarization fluctuation TACF was achieved by minimizing this quantity. The goal of this was to minimize the contribution of the intensity TACF which may be affected by QD blinking to the polarization fluctuation TACF, thereby eliminating the possible complication of QD blinking feed-through into the polarization TACF. Thus this procedure was intended to minimize the correlation between the polarization fluctuation TACF and intensity TACF. Because G_{dd} is typically small relative to G_{ss} and polarizations, as the ratio of these two quantities are smaller still, multiplying the sum autocorrelation by the product of the average polarizations makes this term more comparable in magnitude to the difference autocorrelation in

the first term. The chi-squared χ^2 was defined as the square of this quantity. Because there were different uncertainties associated with each point, we accounted for this by applying weights to each χ^2 point commensurate with their associated uncertainties. We note that although this weighting scheme is not entirely arbitrary and is necessary to account for these uncertainties, this weighting is a lower-order phenomena and, so long as it is appropriately chosen to model the correct trend in uncertainties, shouldn't artifactually hide information present in the polarization fluctuation TACF or conversely create information that was not originally there to begin with even if not derived from first principles. The weighting factor for a χ^2 point W is defined as the reciprocal of the variance σ^2 of that point:

$$W_i = \frac{1}{\sigma_i^2} \quad (3.25)$$

Since these two quantities are inversely related, this means that points with greater uncertainties/variance would be weighted less during the optimization of the polarization fluctuation TACF. We sought to select values for the parameters g , a , and b which would satisfactorily minimize the average weighted chi-squared:

$$\overline{\chi^2}_{wtd} = \frac{\sum_{i=1}^n W_i \chi_i^2}{\sum_{i=1}^n W_i} \quad (3.26)$$

Typical constraints used in the optimization were values of -0.9 to 0.9 for g (corresponding to values of 0.05-19 for g_{trad} , 0-0.005 for a , and 0-0.005 for b). The value of z was fixed at 1 to give a polarization instead of an apparent anisotropy.

After optimizing the above parameters and calculating point-by-point the corrected polarization fluctuation TACF, a single exponential decay was fitted to the points:

$$G_{pp} = G(0)e^{-t/\tau_c} + G(\infty) + \overline{p_1 p_2} \quad (3.27)$$

where t is the time, $G(0)$ is the initial amplitude, τ_c is the correlation time, and the base factor $G(\infty)$ is the limiting polarization (otherwise known as the residual polarization). We note that the average of the product of p_1 and p_2 can be approximated by the product of their averages because the correlation is weak. The square of the deviations of the fitting model from the experimental data d^2 was calculated for each point and for each point a weighted chi-squared using weights w was calculated as follows:

$$\chi_{wtd}^2 = w \times d^2 \quad (3.28)$$

We use w here to designate the weighting factor for each fitted point in the polarization fluctuation TACF and to differentiate this quantity from the weighting factor W used previously which refers to the weights applied to each point when we optimized the polarization fluctuation TACF. This weighting factor was applied to account for the greater scatter/uncertainty in early points of the polarization fluctuation TACF relative to the latter. Because the intervals between

correlation times are much smaller at early times due to the pseudo-logarithmic spacing of correlation times, early channels may contain information resulting from less than an average of one photon per point which accounts for the greater uncertainty in these early bins compared to the latter. Therefore it was necessary to devise a weighting scheme that would give small weights to early points in the correlation curve to compensate for the greater uncertainty associated with these points and progressively larger weights to later points which have smaller uncertainties associated with them. Given the inverse relationship between the error associated with a point and how heavily we wished to weight it, the weight for a point was simply calculated as the reciprocal of the variance σ^2 of the polarization fluctuation TACF. Thus, in order to determine the weights in a rigorous fashion, it was necessary to determine an appropriate formula for the variance of the anisotropy fluctuation TACF. Then the weight was just the reciprocal of this quantity. Calculating the variance of the polarization fluctuation TACF requires that we calculate the variance associated with the product of the deviations of the polarization $r(t)$ from the mean polarization $\mu_{p(t)}$ and the deviations of the time-shifted polarization $p(t+\tau)$ from the mean time-shifted polarization $\mu_{p(t+\tau)}$. For brevity, we will use the following notation for these quantities:

$$\Delta p_1 = p(t) - \mu_{p(t)} \quad (3.29)$$

$$\Delta p_2 = p(t + \tau) - \mu_{p(t+\tau)} \quad (3.30)$$

The autocorrelation of these fluctuations is also easily recognizable as simply the covariance of these two quantities, and thus the problem essentially amounts to calculating the variance of the covariance of these two quantities. Framing this problem as the variance of the product of these

quantities, the following formula for the variance of the product of two correlated variables now applies:

$$\begin{aligned} \sigma_{\Delta p_1 \Delta p_2}^2 &= (\mu_{\Delta p_1}^2 + \sigma_{\Delta p_1}^2)(\mu_{\Delta p_2}^2 + \sigma_{\Delta p_2}^2) \\ &\quad + \sigma_{(\Delta p_1)^2, (\Delta p_2)^2} \\ &\quad - (\mu_{\Delta p_1} \mu_{\Delta p_2} + \sigma_{\Delta p_1, \Delta p_2})^2 \end{aligned} \quad (3.31)$$

If we assume Δp_1 and Δp_2 are independent (*i.e.* uncorrelated), then both the covariance and the covariance of the squares are equal to zero and we obtain the familiar Goodman's expression [7] for the variance of the product of two independent variables:

$$\sigma_{\Delta p_1 \Delta p_2}^2 = \sigma_{\Delta p_1}^2 \sigma_{\Delta p_2}^2 + \sigma_{\Delta p_1}^2 \mu_{\Delta p_2}^2 + \mu_{\Delta p_1}^2 \sigma_{\Delta p_2}^2 \quad (3.32)$$

Because these are averages of fluctuations/deviations from a mean and by definition the first order central moment is zero, the following stipulations apply:

$$\mu_{\Delta p_1} = 0 \quad (3.33)$$

$$\mu_{\Delta p_2} = 0 \quad (3.34)$$

Thus equation 3.34 simplifies to:

$$\sigma_{\Delta p_1 \Delta p_2}^2 = \sigma_{\Delta p_1}^2 \sigma_{\Delta p_2}^2 \quad (3.35)$$

Since this is an autocorrelation of these fluctuations, Δp_1 is similar to Δp_2 and thus we obtain the simple result:

$$\sigma_{\Delta p_1 \Delta p_2}^2 = \sigma_{\Delta p}^4 \quad (3.36)$$

These early points are necessarily modeled by Poisson statistics. A statistical characteristic intrinsic to Poisson distributions is that the standard deviation is equal to the square root of the mean. Thus the variance of the polarization fluctuations is equivalent to the mean of the distribution:

$$\sigma_{\Delta p}^2 = \frac{\mu_{\Delta p}}{n-1} \quad (3.37)$$

For later bins, which are larger and the number of photons per bin greater than one, the uncertainty can be modeled with simple Gaussian statistics.

An exact expression for the variance of a polarization calculated from Poisson variables can be derived from first principles. In practice, to determine the variance of a point of a polarization fluctuation, we used the following approximation:

$$\begin{aligned}\sigma_{\Delta p}^2 &= e^{\frac{x}{1+x^2}} \quad \text{where } x = e^{2^m(p_v+p_h)} \\ &= \frac{e^{2^m(p_v+p_h)}}{1 + e^{2^{m+1}(p_v+p_h)}}\end{aligned}\tag{3.38}$$

Here, x is the combined counts in the vertical and horizontal channels per bin for a given time regime. The counts per clock in the vertical channel p_v and the counts per clock for the horizontal channel p_h are calculated by multiplying the respective average photon count rate per second for the experiment (I_v and I_h) by the fundamental clock time in seconds (1.7×10^{-9} s). These quantities are then summed to give the combined counts per clock for the vertical and horizontal channels. Recognizing the relationship between the spacing of the pseudo-logarithmically-spaced bins, an adjustment can be made to obtain the combined counts per bin. The combined counts per bin is calculated on a given time regime is calculated by multiplying the combined counts per clock for the two channels p_v+p_h by 2^m where m is the binning index introduced previously. For example, the first eight bins have a binning index of $m=10$ so the number of clocks per bin is 2^{10} . Therefore the combined counts per clock is multiplied by 1024 to obtain the combined counts for these first bins. The next eight bins have a binning index of $m=11$, so the counts per clock is multiplied by 2048 to obtain the combined counts for these bins.

A property of the function in Equation 3.38 is that when combined count rate per bin x is much less than one, the x^2 in the denominator becomes very small compared to 1 and the standard deviation can be approximated by $x^{1/2}$. When x is large (much greater than one), the denominator can be approximated by x^2 and the function can be approximated by $x^{-1/2}$. Then this variance is squared to give the variance of the autocorrelation and the reciprocal of this gives the weight for each point. This makes for a good approximation both of the early points best

modeled by Poissonian statistics with the weights decreasing slightly up to the bin in which there is one count per bin/ $\lambda=1$ and then increasing when there is more than one count per bin where the function is best approximated by Gaussian statistics. The variance of the polarization fluctuation TACF was then calculated as the square of this quantity in accordance with equation 3.36, and the weight for a point used for the fitting algorithm was calculated as the reciprocal of this quantity. Once the appropriate weighting was applied to each bin, the average weighted chi-squared was minimized:

$$\overline{\chi_{wtd}^2} = \frac{\sum_{i=1}^n w_i \chi_i^2}{\sum_{i=1}^n w_i} \quad (3.39)$$

The reduced chi-squared was calculated as:

$$\chi_{red}^2 = \frac{\overline{\chi_{wtd}^2}}{n-1} \quad (3.40)$$

We sought to select parameters for the initial TACF amplitude $G_p(0)$, the correlation time τ_c , and the base term $G(\infty)$ that minimized the value of this reduced chi-squared.

3.4. Results and Discussion

3.4.1. Sample Data from TTSPC Experiments

Figure 3.5 shows intensities collected in vertically- and horizontally-polarized channels. QD blinking is responsible for the large fluctuations observed in these traces. The transition of a QD to an “off” state is accompanied by a corresponding decrease in the intensity to the detector dark count rate. Figure 3.6 shows a sample sum/intensity autocorrelation curve for a single cell-bound QD treated with IgE. In this trace, it is observed that the intensity TACF undergoes a steep initial decrease in the first microsecond or so followed by a more gradual decrease thereafter, seeming to asymptotically approach/stabilize at a value of ~ 3000 . Because the intensity function is defined as the the sum of the intensities in the vertical and horizontal orientations, its value is independent of orientation and therefore the decay is not attributable to molecular rotation. The decrease is not completely monotonic, with small fluctuations/noise throughout. These fluctuations are small and not between a zero and non-zero value. This is what would be expected if this phenomenon represented blinking since this is for a single QD which can only assume one of two states (*i.e.* an “on” or “off” state) at any given time. If this was an average of multiple QDs/an ensemble measurement, then blinking would likely not manifest as fluctuations between a zero and non-zero value, because the likelihood that all QDs would coincidentally be in an “off” state at the same corresponding delay times is near negligible/infinitesimally small. Instead, what would be observed would be smaller fluctuations corresponding to a slightly varying proportion of QDs in the “off” state relative to the “on” state at any given time. Comparing to what we observed with our imaging approach, we see the intensity TACF for a single QD yielded results more like what we expected with this imaging approach, with the curve decaying and eventually stabilizing at around zero, but with fluctuations around zero reflecting the transition of the QD from an “on” state to an “off” state.

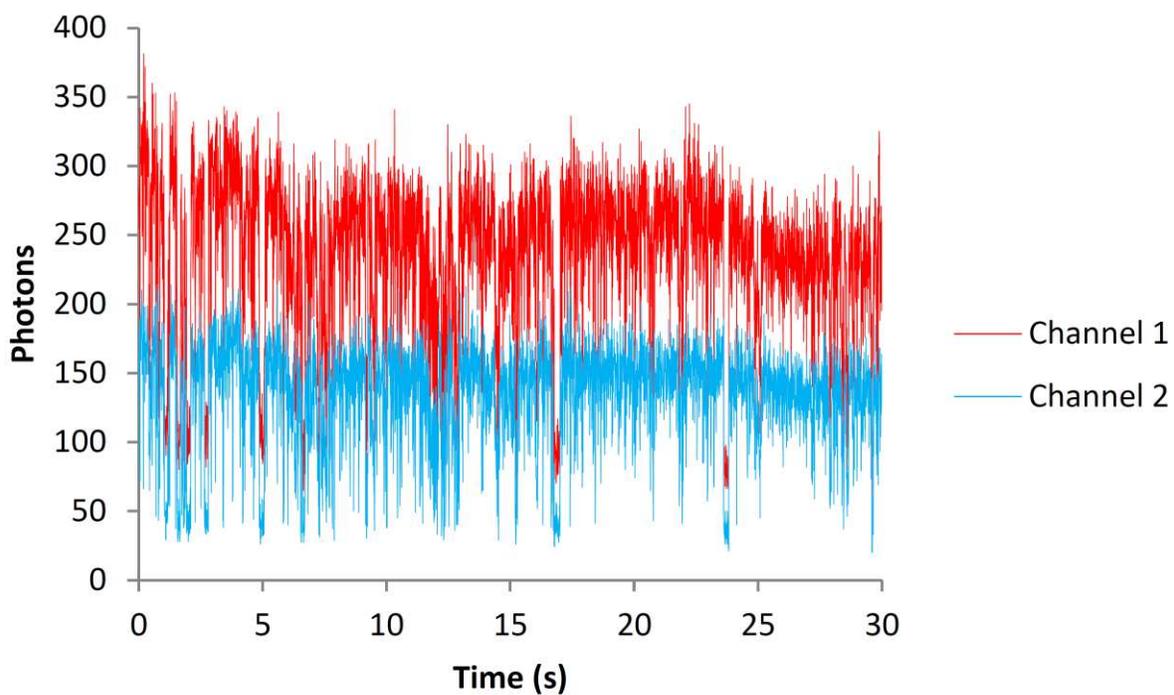
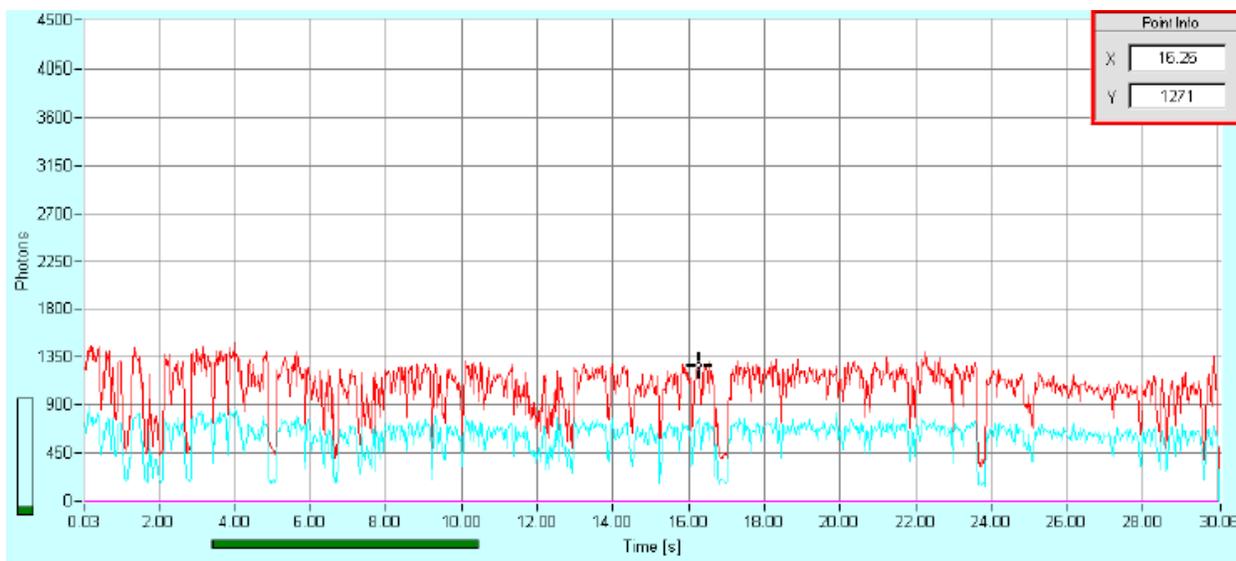


Figure 3.5: Intensities for vertically- and horizontally-polarized channels for IgE-treated RBL-2H3 cells expressing QD605-labeled Fc ϵ RI, channel 1, red, and channel 2, blue, respectively. Observation of these traces reveals large intensity fluctuations over the 30 second run time which are attributable to QD blinking.

Intensity Autocorrelation Function

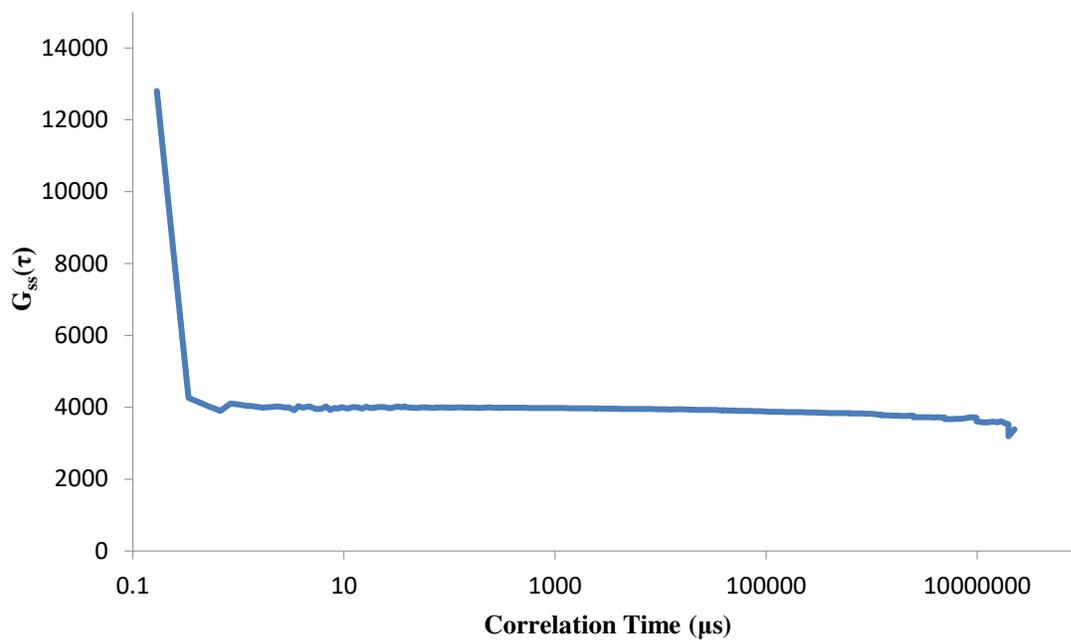


Figure 3.6: Intensity autocorrelation function for cell-bound QD 655 on RBL-2H3 cells treated with IgE. Data was obtained from TTSPC experiments.

Figure 3.7 shows the difference autocorrelation function for this same IgE-treated QD. What is immediately apparent upon comparison of this graph to the respective intensity TACF for this QD are the much smaller magnitudes for the values of the difference TACF relative to those of the intensity TACF. This is because, while the sum of the vertical and horizontal intensities may be a large value, if these respective quantities are similar in value, then the difference between them would be very small. This also means that the *relative* errors associated with each point in the difference TACF are much greater than they are in the intensity TACF. The graph showed an initial steep decay corresponding to the rotation of the QD from the vertical orientation into the horizontal orientation. This would cause a decrease in the former quantity and an increase in the latter, resulting in a smaller difference between the two values.

Figure 3.8 shows the polarization autocorrelation function, defined as the quotient of the difference TACF and the sum/intensity TACF. Because the difference TACF is very small relative to the intensity TACF and the polarization is the ratio of these two quantities, these values are overall very small and even smaller than those associated with the difference TACF. Because the difference TACF initially decays faster than the intensity TACF and then both functions eventually stabilize at some constant, non-zero value, the polarization TACF undergoes a rapid initial decrease before stabilizing at a value of ~ 0.005 , the ratio of these two constant values.

Figure 3.9 shows the variance and corresponding weighting functions used to optimize the polarization fluctuation TACF. A satisfactory model for the variance associated with a point in the polarization fluctuation TACF should be dependent on the binning level (*i.e.* the space

Difference Autocorrelation Function

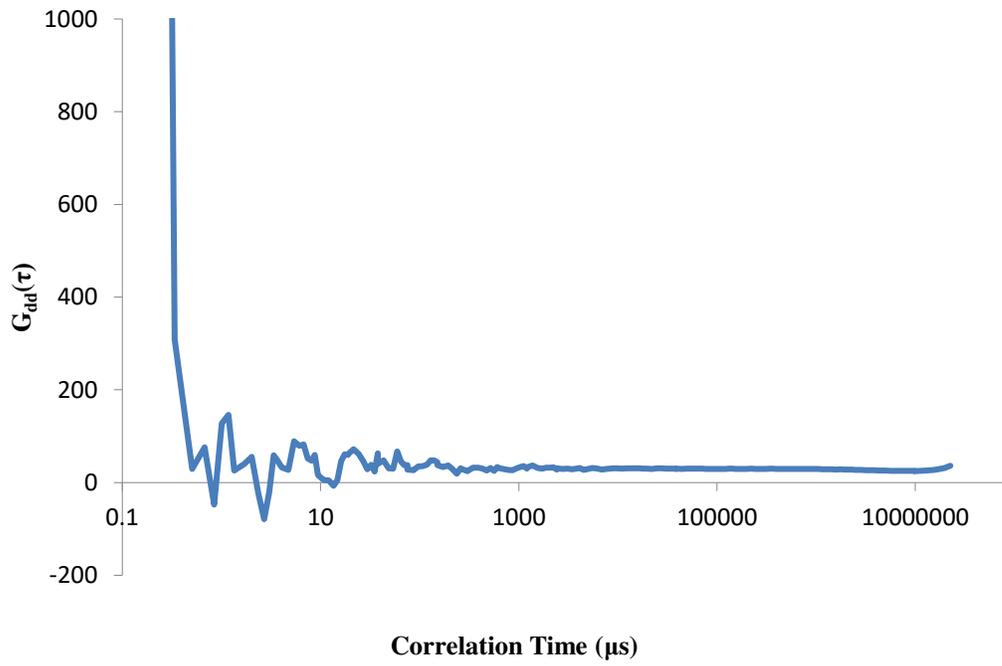


Figure 3.7: Difference autocorrelation function for cell-bound QD 655 on RBL-2H3 cells treated with IgE. Data was obtained from TTSPC experiments.

Polarization Autocorrelation Function

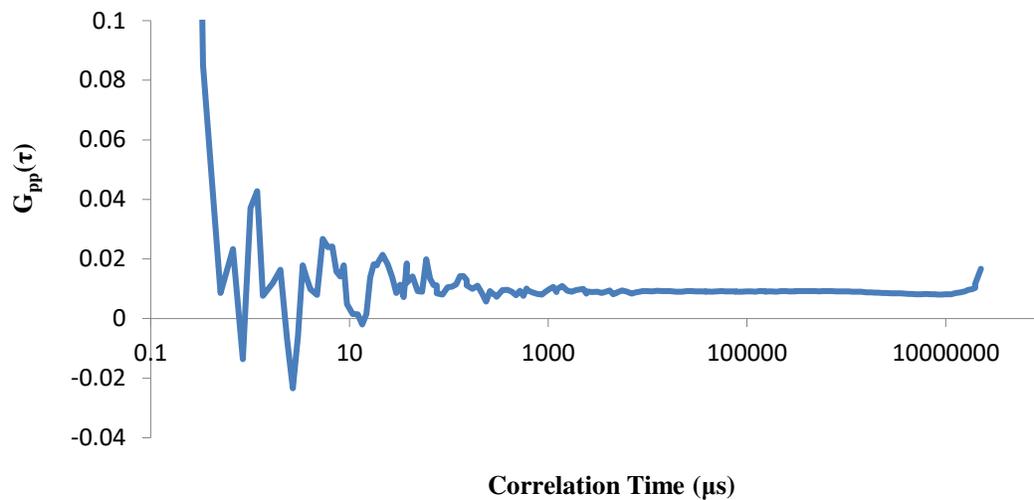


Figure 3.8: Polarization autocorrelation function for cell-bound QD 655 on a RBL-2H3 cell treated with IgE. This data was obtained from a TTSPC experiment.

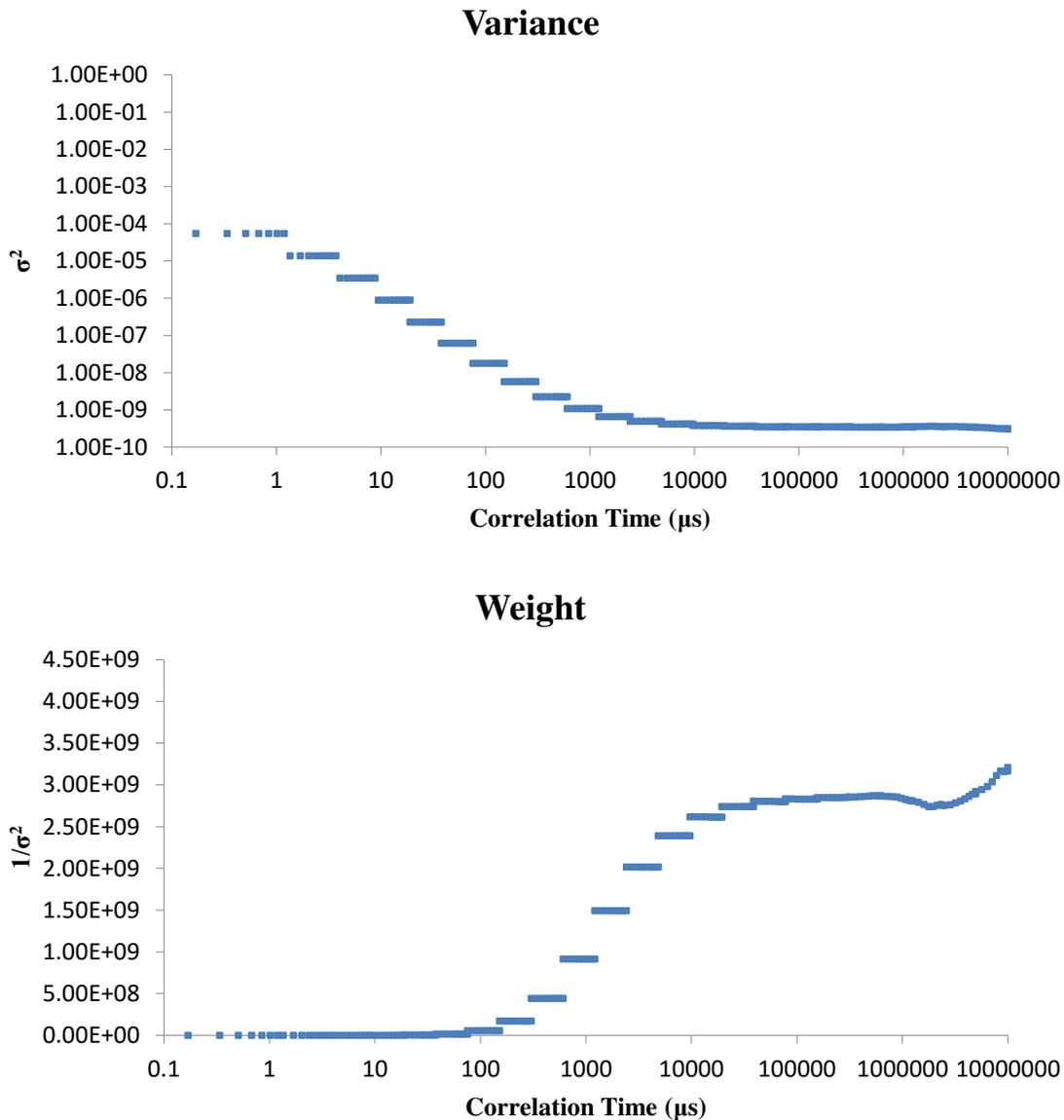


Figure 3.9: Variance function and corresponding weighting function used for optimization of polarization fluctuation TACF. The weight was calculated as the reciprocal of the variance. Data was obtained from TTSPC experiments.

between the points) and should therefore meet two criteria. Firstly, because the bin widths become larger at longer correlation times and the relative uncertainty of any point in the curve becomes smaller as these bin widths become wider, the variance function should display a progressive pointwise decrease at larger correlation times. The exception to this is for a set of points on a given binning regime (*i.e.* with equal spacing between them). In this case, the variance function should remain constant. It is apparent upon inspection of the variance function in the top panel of Figure 3.9 that this function possesses both of these characteristics. The variance decreases with larger bin widths. The variance also remains constant at a given binning level. For instance, for the first eight points spaced by $\sim 0.17 \mu\text{s}$, we observe that the variance remains constant. But for the second set of eight points, this spacing doubles to $\sim 0.34 \mu\text{s}$ and the variance decreases as should be the case given the larger bin width. The variance remains constant for these next eight points until it once again doubles for the next set of points and so on. Thus the variance function appears as a stepwise function with each step corresponding to a particular binning level. Each set of eight points belonging to a binning level are evenly spaced in time, though on the logarithmic scale this appears as a set of points that get closer together with increasing correlation times. The weight function (see the bottom panel of Figure 3.9), inversely related to the variance function as simply its reciprocal, undergoes a progressive increase with longer correlation times and larger bin widths. Applying this weight in the optimization procedure compensates for the greater uncertainty in earlier points with smaller bin widths by weighting them to a lesser degree than those at latter times.

Figure 3.10 displays the points optimized polarization fluctuation TACF along with a trendline giving the moving average calculated with a period of eight. This particular QD gave a

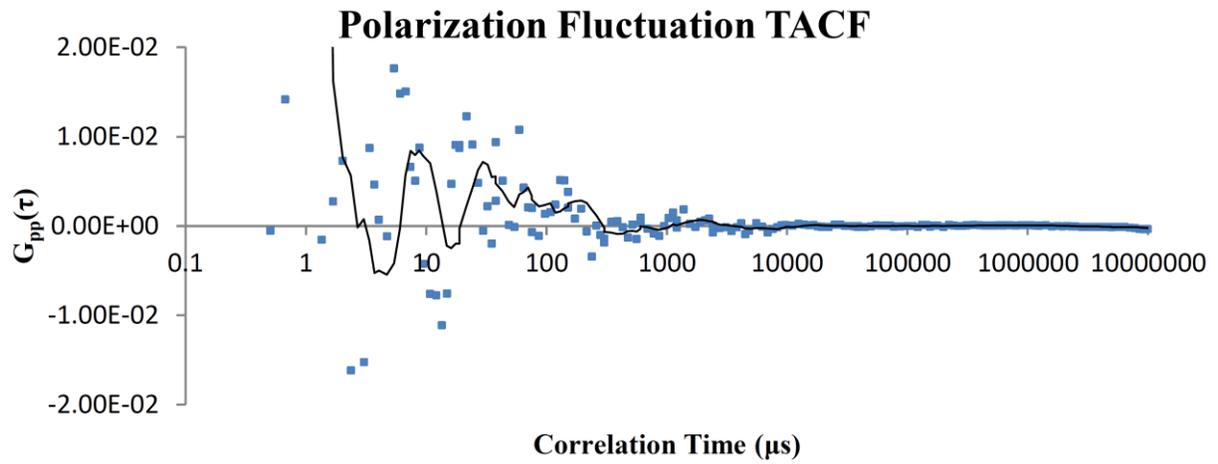


Figure 3.10: Polarization fluctuation TACF for cell-bound QD 655 on RBL-2H3 cells treated with IgE. The graph shows the points with a trend line giving the moving/cumulative average with a period of eight. Data was obtained from a TTSPC experiment.

RCT of 96 μs and a $G_p(0)$ of 0.0044.

Figure 3.11 shows the raw (unweighted) residuals produced by the polarization fluctuation TACF optimization procedure (where the errors χ are defined as a difference between the difference autocorrelation G_{dd} and the product of G_{ss} and $\langle p_{1c} \rangle \langle p_{2c} \rangle$) as well as the weighted residuals and weighted chi-squared ($\chi \times W^{1/2}$ and $\chi^2 \times W$ respectively). Upon inspection of the raw residuals, it is observed that they display increased scatter for early points, commensurate with their greater associated uncertainties. This feature disappears in the weighted residuals and the residuals appear flat and even throughout, suggesting that our weighting scheme accomplished what it was intended for. These weighted residuals show both the direction (*i.e.* positive if $G_{dd} > G_{ss} \times r_1 r_2$ and negative if $G_{dd} < G_{ss} \times r_1 r_2$) and magnitude of the deviations. The weighted chi-squared values, calculated as simply the square of the weighted residuals, show the absolute magnitude of the deviations and are always positive. Optimization involved selecting the parameters a , b , and g that satisfactorily gave a minimum in the chi-squared hypersurface. For this particular QD, the average weighted chi-squared was 1.51.

After successful optimization of the polarization fluctuation TACF, the resulting data points were fit to a one-exponential decay. Figure 3.12 shows this model fitted to the polarization fluctuation TACF data points. This particular QD gave a RCT of 96 μs and a $G_p(0)$ of 0.0044. Also displayed in the bottom two panels are the residual function showing the error between the raw data points and the fitted model as well as the weighted Chi-squared.

3.4.2. Comparing time-tagged single photon counting data for Quantum Dot 605 and 655

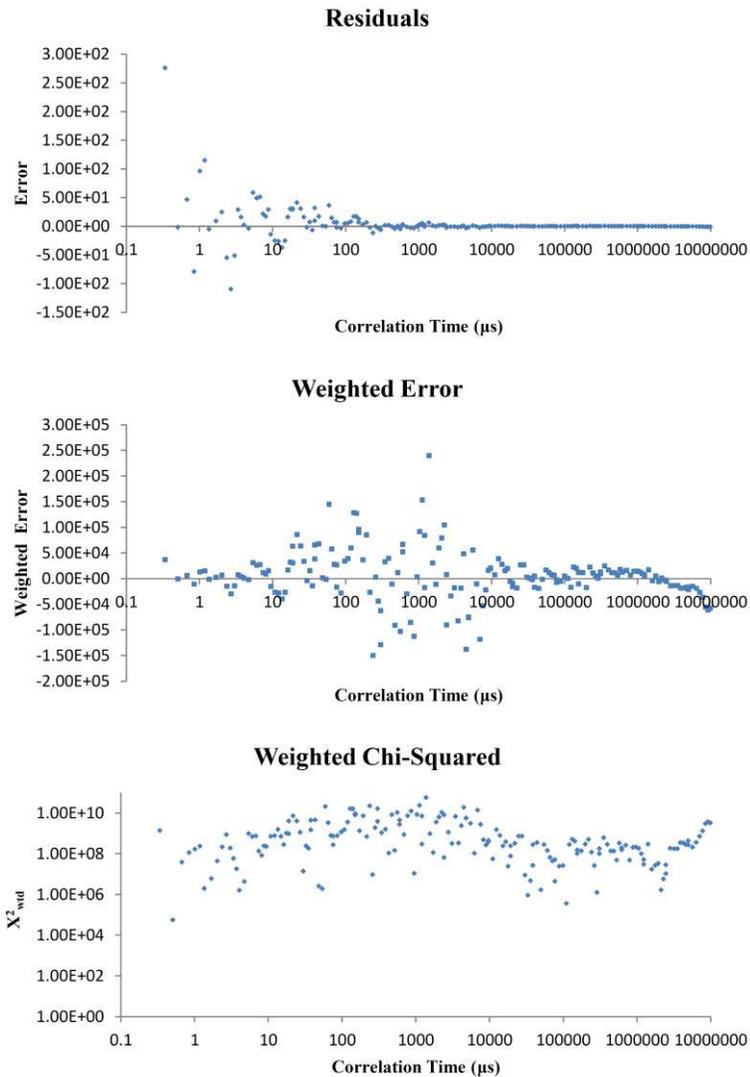


Figure 3.11: Unweighted and weighted residuals/error functions generated during optimization of polarization fluctuation TACF. The top panel shows the raw errors, defined as the difference between the difference autocorrelation and the product of the intensity autocorrelation and $\langle p_1 \rangle \langle p_2 \rangle$. The middle panel shows the error multiplied by the square root of the weight and thus shows both the direction and magnitude of the deviations whereas the bottom panel showing the weighted chi-squared values gives the square of this and thus only shows the absolute magnitude of the deviations. Optimization of the polarization fluctuation TACF involved minimizing the weighted average of these deviations. Data was obtained from a TTSPC experiment.

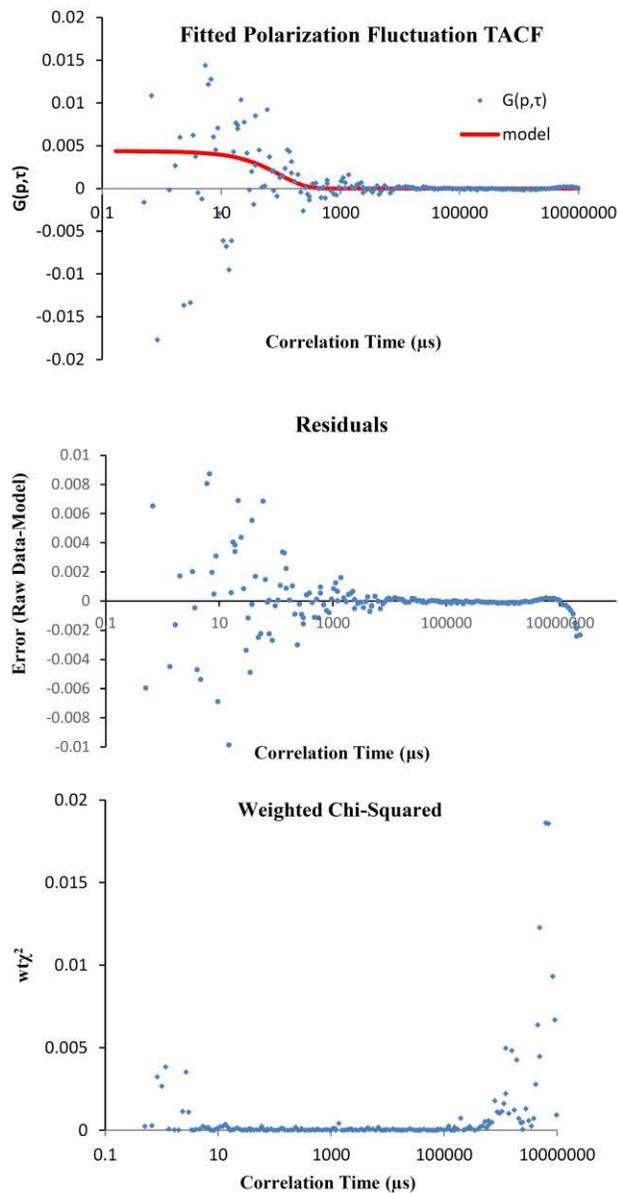


Figure 3.12: Polarization fluctuation TACF for cell-bound QD 655 on RBL-2H3 cells treated with IgE. The top panel shows the model fitted to the raw data points. This QD had a rotational correlation time of $\sim 96 \mu\text{s}$ and an amplitude of 0.0044. The middle panel shows the difference of the model from the fitted points. The bottom panel shows the weighted residuals.

For all treatment groups except the paraformaldehyde-treated group for which we conducted experiments using only QD605 probes, there were two subgroups of experiments. In one subgroup the IgE was labeled with a QD605 probe, while in the other subgroup the IgE was labeled with a QD655 probe. We hypothesized that any difference in results between these two groups would not be attributable to a difference in the actual rotation of receptors that have been labeled with different probes. This is because, since both the IgE ligand and QD probe are situated outside the membrane in the extracellular aqueous environment, we do not expect the IgE, let alone the QD probe, to hinder receptor rotation to a considerable degree. An extensive theoretical treatment of this matter can be found in Chapter II where we concluded that effectively all the rotational restraint came from the membrane-embedded receptor. Therefore we hypothesize that any difference in these two groups would be attributable to differences in the optical properties of these two probes and not differences in rotation.

The most obvious difference in the optical properties of these two probes is the wavelength at which they emit light. Though both QD605 and QD655 have a wavelength of maximum absorbance around 456 nm, as their names suggest, they differ in their wavelength of maximum emission (605 nm for QD605 and 655 nm for QD655). Another difference of potential significance between QD605 and QD655 are their differing aspect ratios.

Table 3.1 shows the weighted geometric average rotational correlation times and associated statistics including the weighted geometric standard deviations and standard errors along with the total number of experiments in a subgroup and the number of experiments selected for treatment for subgroups treated with QD605- and QD655- labeled IgE. Table 3.2

Table 3.1: Comparison of rotational correlation times and associated statistics of QD 605 and QD 655 subgroups for all five treatment groups. These data were obtained from TTSPC experiments.

Group	QD emission wavelength (nm)	1st reagent	2nd reagent	ln(Wtd. Geo. Avg. RCT)	Wtd. Geo. Avg. RCT (μ s)	Wtd. Geo. SD RCT (μ s)	Std. Err. RCT (μ s)	Tot. QD	QD Selected
QD 605 IgE	605	IgE	-	8.1	3147.6	-	-	5	1
QD 655 IgE	655	IgE	-	5.4	224.7	518.8	366.8	5	2
QD 605 DNP-BSA	605	IgE	DNP-BSA	8.1	3303.2	-	-	5	1
QD 655 DNP-BSA	655	IgE	DNP-BSA	-	-	-	-	5	0
QD 605 Pf	605	IgE	Pf	-0.5	0.6	1.4	0.8	5	3
QD 655 Pf	655	IgE	Pf	-	-	-	-	-	-
QD 605 M β CD	605	IgE	M β CD	0.2	1.2	6.7	3.4	5	4
QD 655 M β CD	655	IgE	M β CD	-0.8	0.4	1.7	1.0	5	3
QD 605 CytoD	605	IgE	CytoD	3.9	50.0	-	-	5	1
QD 655 CytoD	655	IgE	CytoD	2.7	14.3	206.7	43.1	41	23

Table 3.2: Comparison of polarization TACF amplitudes and associated statistics of QD 605 and QD 655 subgroups for all five treatment groups. These data were obtained from TTSPC experiments.

Group	QD emission wavelength (nm)	1st reagent	2nd reagent	Wtd. Arith. Avg. $G_p(0)$	SD $G_p(0)$	SE $G_p(0)$	Tot. QD	QD Selected
QD 605 IgE	605	IgE	-	0.00356	-	-	5	1
QD 655 IgE	655		-	0.00251	0.00261	0.00185	5	2
QD 605 DNP-BSA	605	IgE	DNP-BSA	0.00452	-	-	5	1
QD 655 DNP-BSA	655		DNP-BSA	-	-	-	5	0
QD 605 Pf	605	IgE	Pf	0.38310	0.53775	0.31047	5	3
QD 655 Pf	655		Pf	-	-	-	-	-
QD 605 M β CD	605	IgE	M β CD	0.21866	0.23153	0.11577	5	4
QD 655 M β CD	655		M β CD	1.00000	0.00000	0.00000	5	3
QD 605 CytoD	605	IgE	CytoD	0.00500	-	-	5	1
QD 655 CytoD	655		CytoD	0.02057	0.02928	0.00611	41	23

shows the weighted average arithmetic amplitudes of the polarization fluctuation TACFs and associated statistics including the weighted arithmetic standard deviation and standard errors along with the total number of experiments in a subgroup and the number of experiments selected for treatment for these respective subgroups. Unfortunately, many QDs exhibited unreasonable values for their rotational correlation times and/or amplitudes and were therefore omitted in calculations of averages and associated statistics. The majority of these outliers fell into two categories: QDs that exhibited very long rotational correlation times and small polarization fluctuation TACFs, and QDs that exhibited very short correlation times and large $G_p(0)$ s. We speculated that both of these cases could indicate immobile QDs. In the former case, a long rotational correlation time could be indicative of a QD which does not decay on the timescale being observed. Such a QD could either decay on a much longer timescale yet still appear immobile on this shorter timescale (indicating slow, hindered rotation) or could be completely immobile. Perhaps less intuitive is that, in the latter case as we have discussed previously, a short rotational correlation time is not necessarily indicative of fast rotation. A very short, near-zero rotational correlation time could alternatively be an indicator of receptor immobility. The accompanying large $G_p(0)$ can be attributed to shot noise in the first point resulting from the correlation of each point with itself. A third category of outliers was designated for those QDs that exhibited rotational correlation times and amplitudes that could reasonably be interpreted to represent actual rotation, yet, given the treatment group had enough QDs selected to enable us to observe a discernable trend with sufficient statistical confidence, exhibited behavior that differed from the majority of the other QDs included in final analysis (i.e. was not reproducible).

For many of the subgroups, only 2, 1, or even 0 QDs exhibited behavior that could be interpreted as actual rotation. For instance, for IgE-treated cells, only 1/5 and 2/5 of the QDs exhibited such behavior for the QD605 and QD655 subgroups, respectively. For the DNP-BSA-treated group, these numbers were 1/5 and 0/5 respectively. For the paraformaldehyde-treated group, all experiments were conducted using QD605 so a comparison could not be made for this treatment group. Out of these 1/5 QDs displayed rotation. For the M β CD-treated group, 1/5 and 0/5 QDs were selected for the QD605 and QD655 subgroups respectively. For the cytochalasin D group, 1/5 QDs were selected in the QD605 group. Because so few QDs in the QD605 or QD655 subgroups for each treatment exhibited behavior that could be attributed to actual rotation and most QDs appeared to be immobile, any differences between groups lack statistical significance and therefore it is difficult to draw any conclusions about differences between the QD605 and QD655 subgroups.

3.4.3. Effects of cell treatments as observed by time-tagged single photon counting

Table 3.3 shows compiled results for the weighted geometric average rotational correlation times and associated statistics obtained for the five treatment groups. Table 3.4 shows compiled results for the weighted arithmetic average polarization fluctuation TACF amplitudes and associated statistics for these same five treatment groups. As observed in the “QDs Selected” column, relatively few QDs were chosen for calculation of final statistics. This is because many QDs exhibited unreasonable values for their rotational correlation times and/or amplitudes. As mentioned in the previous section, the majority of these outliers fell into two categories: QDs that exhibited very long rotational correlation times and small polarization fluctuation TACFs

Table 3.3: Comparison of rotational correlation times and associated statistics for all five treatment groups. These data were obtained from TTSPC experiments.

Group	1st reagent	2nd reagent	ln(Wtd. Geo. Avg. RCT)	Wtd. Geo. Avg. RCT (μ s)	SD RCT (μ s)	Std. Err. RCT (μ s)	Tot. QD	QD Selected
IgE	IgE	-	6.3	541.7	2553.9	1474.5	10	3
DNP-BSA	IgE	DNP-BSA	8.1	3303.2	-	-	10	1
Pf	IgE	Pf	-0.5	0.6	1.4	0.8	5	3
M β CD	IgE	M β CD	-0.2	0.8	3.5	1.3	10	7
CytoD	IgE	CytoD	2.7	15.1	206.8	42.2	46	24

Table 3.4: Comparison of polarization TACF amplitudes and associated statistics for all five treatment groups. These data were obtained from TTSPC experiments.

Group	1st reagent	2nd reagent	Wtd. Arith. Avg. $G_p(0)$	SD $G_p(0)$	SE $G_p(0)$	Total QD	QD Selected
IgE	IgE	-	0.00286	0.00194	0.00112	10	3
DNP-BSA	IgE	DNP-BSA	0.00452	-	-	10	1
Pf	IgE	Pf	0.38310	0.53775	0.31047	5	3
M β CD	IgE	M β CD	0.55352	0.44859	0.16955	10	7
CytoD	IgE	CytoD	0.01992	0.02881	0.00588	46	24

and QDs that exhibited very short near-zero rotational correlation times and large polarization fluctuation TACFs. A third category of outliers was reserved for those QDs exhibiting rotational correlation times which could reasonably be interpreted as representing rotation, yet displayed behavior that was not typical of the group (i.e. was not reproducible).

3.4.3.1. Immunoglobulin E Only

Because, unlike our imaging approach, our photon-counting approach has the capability of accessing the microsecond timescale, we hoped this technique would enable us to observe the hydrodynamic rotation of the receptor. As previously mentioned, this rotational correlation time has been measured to be 27 μs at 37 $^{\circ}\text{C}$ by time-resolved phosphorescence anisotropy [1]. Assuming a photon count rate of at least $\sim 37,000$ photons/s, such a timescale should be accessible to our photon-counting approach. At 4 $^{\circ}\text{C}$, this slows to 82 μs , requiring only $\sim 12,000$ photons/s [3]. At 25 $^{\circ}\text{C}$ as conducted in this study, this value is ~ 40 μs , requiring 25,000 photons/s to yield one photon per correlation time [2]. We expect less variability in rotational correlation time from dot-to-dot and hence a smaller standard deviation than we observed with the imaging approach. This is because, while with the imaging group the variability may have been due to receptors occupying lipid domains of varying sizes, we do not expect to observe such long-lived rotation with our photon-counting approach and instead expect to observe the hydrodynamic rotation of the IgE-bound receptor. Because no polyvalent antigen is present to crosslink the IgE-bound receptor, we do not expect higher molecular weight receptor complexes.

We obtained a weighted geometric average rotational correlation time of 542 μs and an accompanying standard error of $\pm 1475 \mu\text{s}$ for IgE-treated cells (see Table 3.3). We obtained a weighted arithmetic average polarization fluctuation TACF amplitude of 0.002863 and an accompanying standard deviation of 0.001943 (see Table 3.4). Because the majority of quantum dots exhibited unreasonable values for rotational correlation times and/or amplitudes that could not possibly be attributable to hydrodynamic rotation of the receptor, only three out of ten QDs were ultimately selected to be used for further calculations of averages and associated statistics. Three of these seven excluded QDs had very small correlation times on the order of sub-microsecond to several microseconds and very large polarization fluctuation TACF amplitudes exceeding a value of 0.02. These QDs were considered to be immobile. The rationale for this interpretation has been discussed in previous sections. The other four of these seven excluded QDs had very large correlation times ranging from 200 ms up to 1 s combined with small polarization TACF amplitudes ranging from as small as 0.00002 to as large as 0.00081. These results could also indicate QDs that are completely immobile or, alternately, could indicate hindered rotation with decay occurring on a longer timescale such as would be expected for lipid rafts (we note that in this context, the term “raft” is used as a generic term for cholesterol-rich microdomains and isn’t intended to have the more specific characteristics sometimes associated with the term). This would provide an explanation for the long rotational correlation times, although the smaller-than-expected amplitudes are difficult to explain. Of the three QDs that were selected, one QD in particular, the data for which is shown in section 3.4.2 “Sample data for TTSPC experiments”, gave promising results that suggest that what is being observed here is indeed the hydrodynamic rotation of the receptor. This QD gave a rotational correlation time of 96.4 μs and a polarization fluctuation TACF amplitude of 0.00436. The other two QDs that were

included had rotational correlation times of 524 μ s and 3148 μ s and polarization fluctuation TACF amplitudes of 0.00067 and 0.00356 respectively. Both of these rotational correlation times are orders of magnitude too large to represent hydrodynamic rotation of the receptor, yet could still be interpreted to represent meaningful rotation. Perhaps in these cases, the receptor has become integrated into some larger complex of unknown identity and this is responsible for the larger-than-expected rotational correlation time. The cells were not treated with polyvalent antigen, so no crosslinking of the IgE-bound receptor is expected and this could not be responsible for the inflated rotational correlation times for these two QDs. It is not possible to ascertain the identity of such a binding partner with this technique and thus we can only speculate on what might cause the hindered rotation. This is because this technique is characteristically non-specific. This can be construed as either an advantage or a disadvantage. It can be advantageous when investigating whether unknown interactions with the receptor occur as it does not require labeling of a known binding partner. Thus this technique can be regarded as an exploratory one capable of revealing binding events that would otherwise be overlooked when using a more specific technique such as FRET. However, this can also be considered a limitation because it gives no information on the identity of possible binding partners of the receptor. Thus further studies with alternative methods would be needed to determine the cause of the hindered rotation in these two cases and elucidate the identity of any possible binding partner that could be responsible.

3.4.3.2. Immunoglobulin E and DNP-BSA

We obtained a weighted geometric average rotational correlation time of 3302.2 μs . Because only a single QD was included, no standard error was calculated for DNP-BSA-treated cells (see Table 3.3). We obtained a weighted arithmetic average polarization fluctuation TACF amplitude of 0.004517 (see Table 3.4). As was the case for the IgE-treated group, most QDs either exhibited large correlation times and small amplitudes or small correlation times and large amplitudes, suggestive of immobility or hindered rotation. A total of nine out of ten QDs exhibited this behavior, with seven of these possessing rotational correlation times ranging from the sub- μs timescale to several μs and amplitudes exceeding a value of 0.03. The other two excluded QDs exhibited rotational correlation times in the 100 ms to 1 s range and amplitudes <0.0005 . Because this treatment was expected to result in hindered rotation, it is possible that this large proportion of QDs that are seemingly immobile is legitimately attributable to the crosslinking of receptors by DNP-BSA; however, because a similar proportion of QDs displaying these characteristics were present in other groups including those not expected to hinder rotation to an appreciable extent (*e.g.* IgE), it is possible that this is instead attributable to systematic error in analysis. It is impossible to determine which explanation is correct using solely this method. Our imaging approach shows rotation on a longer timescale, but it was unclear whether this was due to crosslinking or libration of lipid rafts. Furthermore, including a limiting polarization as a fitted parameter in these studies on a longer timescale may reveal a population of receptors displaying absolute immobility that do not decay on any timescale. In any case, we decided to assume the latter and exclude these QDs from further analyses. A single QD exhibited a more reasonable correlation time of 3302.2 μs . Though this is almost a couple orders of magnitude greater than what would be expected for hydrodynamic rotation of the

receptor, given extensive crosslinking of receptor by DNP-BSA, such a rotational correlation time is plausible.

3.4.3.3. Immunoglobulin E and paraformaldehyde

Paraformaldehyde is a fixative agent which is expected to cause membrane protein aggregation via intermolecular crosslinking of these proteins through covalent bonds. We hypothesize that fixation of receptors with paraformaldehyde will significantly slow (if not completely immobilize) receptors. This could possibly manifest as either a very long rotational correlation time or as an increase in the limiting polarization in our experiments. Our group has previously studied the effect of various concentrations of paraformaldehyde on the lateral and rotational dynamics of the MHC class II antigen I-A^d on A20 cells [8]. Lateral and rotational diffusion were measured by fluorescence recovery after photobleaching (FRAP) and time-resolved phosphorescence anisotropy (TPA) using tetramethyl-rhodamine-labeled MKD6 Fab fragments as probes. Together with these two different techniques, it was possible to classify membrane structures into three distinct groups. The first of these groups consisted of those membrane proteins or small complexes which exhibited measurable mobility in TPA measurements suggesting they were of small size. In TPA measurements, any small change in size would appear as an increase in rotational correlation time/decrease in the rotational diffusion constant manifesting as a shift of the decay to the right. This would occur without any increase in the limiting anisotropy which is indicative of the immobile fraction observed by this technique. Such rotational mobility was only observed for untreated cells on the timescale of TPA measurements. A second group was comprised of intermediately-sized and likely higher-order

structures which were immobile in TPA measurements, yet still exhibited measurable lateral mobility as reflected by a lack of change in the observed mobile fraction. This decrease in the lateral diffusion constant was found to be concentration-dependent. This scenario was observed for paraformaldehyde concentrations ranging from 0.2 % to up to 2%. In TPA measurements, what was seen was an increase in the limiting anisotropy on this timescale, reflecting a larger proportion of molecules which were immobile on this timescale. Since these molecules exhibited measurable lateral mobility, this means that the molecules are likely not rotationally immobile in an absolute sense, but rather simply rotate too slowly for this rotation to be measured in TPA experiments. In FRAP measurements, upon increasing paraformaldehyde concentrations, the observed increase in fluorescence during the fluorescence recovery period was more gradual following photobleaching, reflected by a slower fitted lateral diffusion constant [8]. Additionally, the observed mobile fraction, reflected in the ratio of the final fluorescence intensity after recovery to the initial fluorescence intensity before photobleaching, did not change over this range of concentrations. This suggested that these molecules, while immobile on the timescale of TPA measurements, still exhibited measurable lateral mobility. It is worth mentioning that the fact that these molecules were rotationally immobile on the timescale of TPA measurements does not preclude them from being rotationally mobile on a much slower timescale, and, more than this, the fact that these molecules are laterally mobile suggests that they also exhibit some rotation, albeit slow, on a longer timescale. A final group encompassed very large, definitively higher-order structures which both laterally and rotationally immobile as shown by photobleaching and TPA experiments respectively.

We obtained a weighted geometric average rotational correlation time of 0.6 μs and an accompanying standard error of $\pm 0.8 \mu\text{s}$ for paraformaldehyde-treated cells (see Table 10). We obtained a weighted arithmetic average polarization fluctuation TACF amplitude of 0.3831 and an accompanying standard deviation of ± 0.5377 (see Table 11). Of the five QDs, three exhibited very small rotational correlation times (sub-microsecond to on the order of a few microseconds) and very large amplitudes (>0.01). Of the remaining two, one exhibited a short correlation time of 11.8 μs and an amplitude of 0.00203. Though the amplitude is a reasonable magnitude, in order to get one count per correlation time for a correlation time this short, a count rate of about 85,000 counts/s would be required which is approximately the lower threshold accessible by this technique. For this particular QD, the vertical and horizontal intensities were 167401 and 87226 counts/s respectively. So, while enough counts were acquired in order to observe such a correlation time, it is difficult to explain such a short rotational correlation time. It is possible that there is a population of receptors that were not crosslinked. At 25 $^{\circ}\text{C}$, the hydrodynamic rotation of the receptor has been measured to be $\sim 40 \mu\text{s}$. This value of 11.8 μs , though low, is well within the margin of error and could indeed represent the hydrodynamic rotation of the lone IgE-bound receptor. The other QD had a rotational correlation time of 2041.1 μs and an amplitude of 0.00039. This rotational correlation time suggests hindered rotation, but, for paraformaldehyde, we expected complete immobilization. Furthermore, this amplitude is significantly lower than what we might expect. For these reasons, we believe these two QDs were outliers because they differed from the characteristic combination of short correlation time and long amplitude exhibited by the other three QDs and exhibited behavior that was not suggestive of immobility as we would expect for these paraformaldehyde-treated QDs. For this reason, we excluded these two QDs and only included the three seemingly immobile QDs in

calculations of averages and associated statistics. Since this pattern of short correlation time and long amplitude was also fairly prevalent for many QDs in the other treatment groups, it is difficult to determine whether this is simply a systematic error present throughout our measurements, or if this actually represents the immobilization effects of paraformaldehyde.

3.4.3.4. Immunoglobulin E and methyl- β -cyclodextrin

Predicting the effects of membrane cholesterol depletion by M β CD on the rotational diffusion of the Fc ϵ RI is more difficult than for the previously discussed treatments. Initial considerations might lead one to expect that this treatment might eliminate lipid rafts because cholesterol is an essential constituent of these domains. The logical conclusion following from such an assumption is that the receptor, now liberated from these domains, would transition from an environment of high localized viscosity to one of lower viscosity. The receptor would then be expected to undergo a corresponding increase in both its lateral and rotational diffusion constants. However, as discussed in Chapter II, we have proposed a model in which cholesterol reduces order in lipid rafts. These domains are enriched in saturated phospholipids and sphingolipids which have a high packing efficiency. The enrichment of these domains in cholesterol is theorized to disrupt this tight packing, thereby preventing these domains from solidifying. Therefore we expect depletion of membrane cholesterol by M β CD to increase order in these domains, resulting in hindered lateral and, by extension, rotational diffusion [9]. Indeed, a number of studies have observed decreased lateral diffusion upon treatment with M β CD, consistent with this hypothesis [10-12]. However, this was found to be true for some non-raft-localized membrane proteins as well. This is surprising because, according to this hypothesis,

cholesterol should have the exact opposite effect outside of a lipid raft—that is, it should increase membrane order. Although cholesterol is less abundant outside of lipid rafts, it still has an important role in this bulk membrane environment. This phase tends to be enriched in unsaturated phospholipids which do not pack together tightly and have a large degree of conformational freedom. The rigid, planar steroid ring of cholesterol constrains the number of conformational states that can be assumed by the fatty acid tails of adjacent phospholipids, imparting a more tightly-packed, ordered structure. Thus we would expect depletion of cholesterol with M β CD to decrease membrane order outside of lipid rafts, thereby increasing both lateral and rotational diffusion. Shvartsman *et al.* [13] have reported a similar finding for non-raft-localized mutants of Ras and influenza hemagglutinin. Furthermore, although Shvartsman *et al.* obtained results suggesting that, while M β CD did retard the lateral diffusion of raft-localized Ras and influenza hemagglutinin, compactin, a metabolic inhibitor of cholesterol production, had the opposite effect and α -cyclodextrin (α -CD), a cyclodextrin which nonspecifically depletes membrane cholesterol, also decreased lateral diffusion of these raft-localized proteins. Based on these results, they hypothesized that the decrease in the lateral diffusion constant of the raft-localized proteins induced by M β CD was not attributable to actual cholesterol depletion, but instead attributable to a collateral effect involving the non-specific depletion of other membrane lipids. They argue that, because compactin should specifically target cholesterol and α -CD is not specific for cholesterol and, since the effects of M β CD are opposite that of compactin yet similar to α -CD, M β CD effects on lateral diffusion are independent of its modulation of membrane cholesterol content, and instead attributable to its effects on the membrane content of other lipids. We regard these claims with some skepticism, as M β CD was designed to be much more specific for cholesterol than other cyclodextrins. In

support of this, previous studies have shown that treatment with increasing concentrations of M β CD generate a progressive reduction in levels of membrane cholesterol while simultaneously having an insignificant effect on the total phospholipid content of the membrane [14]. Whether this holds true for other membrane lipids is a topic worthy of further investigation. Regardless of which model is more accurate, we expect the same effect on the lateral and, by extension, rotational diffusion constant of raft-localized Fc ϵ RI upon treatment with M β CD. We expect to observe a decrease in the rotational diffusion constant in these studies.

We obtained a weighted geometric average rotational correlation time of 0.8 μ s and an accompanying standard error of ± 1.3 μ s (see Table 3.3). We obtained a weighted arithmetic average polarization fluctuation TACF amplitude of 0.5535 and an accompanying standard deviation of ± 0.4486 (see Table 3.4). In selecting data for inclusion in calculations of averages and associated statistics, we looked for trends in the distribution of extracted parameters. Since the majority (seven out of ten) of the QDs displayed very short correlation times (sub-microsecond to on the order of several microseconds) along with large amplitudes (>0.009), we interpreted this to represent the most probable outcome of these experiments. A total of three out of the ten QDs were excluded from final analysis. Of these outliers, two displayed very long correlation times on the order of several hundred ms) and small amplitudes (<0.0005). For the other outlier, the parameters did not change from the incumbent values (selected by visual inspection although this was admittedly difficult given the large amount of scattering/noise in early points) during the fitting routine, perhaps because the algorithm did not explore the full space of possible optimization solutions and this initial value was the best local solution that could be found. The large proportion of QDs exhibiting very short correlation times and large

amplitudes could be explained in two ways. The first (and most likely) explanation is that this is the result of a systematic error. There was a high prevalence of QDs exhibiting this behavior for all treatments and not just the M β CD-treated cells, suggesting that this effect was independent of treatment. A second possible explanation is that the treatment with M β CD has indeed immobilized a proportion of receptors on the cell surface. Such an effect would be consistent with our hypothesis of cholesterol depletion with M β CD increasing order in lipid rafts, thereby slowing the rotational diffusion of raft-localized Fc ϵ RI. Whether this would produce a modest decrease in diffusion or complete immobilization of the receptor is hard to determine. Neither would be accessible on this timescale, so it would be difficult to differentiate between these two possibilities. The two QDs that displayed long correlation times and small amplitudes could be considered immobile, although this is likely also due to systemic error, as such QDs were also fairly common regardless of treatment.

3.4.3.5. Immunoglobulin E and Cytochalasin D

In a study using time-resolved phosphorescence anisotropy with erythrosine-conjugated IgE conducted by Torigoe *et al.*, the RCT did not significantly decrease upon treatment with cytochalasin D [15]. This lack of effect was observed both for unaggregated receptor and from ligand-aggregated receptors.

We obtained a weighted geometric average rotational correlation time of 15 μ s and an accompanying standard error of 42 μ s for cells treated with cytochalasin D (see Table 3.3). We

obtained a weighted arithmetic average polarization fluctuation TACF amplitude of 0.0199 and an accompanying standard deviation of 0.0288 (see Table 3.4).

3.5. Conclusions

In these experiments, we observed a high proportion of QDs with fast rotational correlation times and large polarization TACF amplitudes. QDs exhibiting this behavior occurred throughout our experiments, independent of treatment and the proportion exhibiting this behavior also remained relatively unchanged regardless of treatment. The magnitude of these rotational correlation times were typically in the range of sub-microsecond to several microseconds. This is well below the rotational correlation time that would be expected for any membrane protein.

We also observed a large fraction of QDs which exhibited the inverse phenomenon of long correlation times and small polarization TACF amplitudes. It is possible that the high prevalence of QDs that exhibited large correlation times and small amplitudes is an artifact of our analysis and does not reflect a population of receptors that is actually absolutely immobile. The model which we fitted to the polarization fluctuation TACF decays was a simple single-exponential and did not include a constant base term to account for a possible limiting polarization. In theory, the polarization fluctuation TACF, which represents the correlation of fluctuations about the mean polarization, should decay to zero. It is possible then, if a three-parameter fit were the more appropriate fitting model for the data, that by fitting to a single-exponential with no base term which decays to zero, the model would at long times decay very slowly in order to minimize the errors between it and the actual data which asymptotically

approaches some non-zero limiting polarization. This would explain the long rotational correlation times.

3.6. References

1. Song, J., et al., *Interactions of the mast cell function-associated antigen with the type I FcEpsilon receptor*. *Molecular Immunology*, 2001. **38**: p. 1315-1321.
2. Zhang, D., et al., *Time-tagged single photon counting examination of rotation of receptor-bound quantum dots*. *Biophysical Society 2019 Abstract*, 2019.
3. Song, J., et al., *The mast cell function-associate antigen and its interactions with the Type I Fcepsilon receptor*. *Biochemistry*, 2002. **41**: p. 881-889.
4. Zhang, D., et al., *Rotation of Single Cell Surface Receptors Examined by Quantum Dot Probes*, in *Membrane Organization and Dynamics*, A. Chattopadhyay, Editor. 2017, Springer International Publishing. p. 287-309.
5. Bohmer, M., et al., *Advanced time-resolved confocal scanning device for ultrasensitive fluorescence detection*. *Rev. Sci. Instrum.*, 2001. **72**: p. 4145-4152.
6. Wahl, M., et al., *Fast calculation of fluorescence correlation data with asynchronous time-correlated single-photon counting*. *Optics Express*, 2003. **11**: p. 3583-3591.
7. Goodman, L., *On the Exact Variance of Products*. *Journal of the American Statistical Association*, 1960. **55**(292): p. 708-713.
8. Barisas, B., et al., *Dynamics of molecules involved in antigen presentation: effects of fixation*. *Molecular Immunology*, 1999. **36**(11-12): p. 701-708.
9. Nishimura, S., et al., *Cholesterol depletion induces solid-like regions in the plasma membrane*. *Biophysical Journal*, 2006. **90**: p. 927-938.

10. Vrljic, M., et al., *Cholesterol depletion suppresses the translational diffusion of class II major histocompatibility complex proteins in the plasma membrane*. Biophysical Journal, 2005. **88**: p. 334-347.
11. Kenworthy, A., et al., *Dynamics of putative raft-associated proteins at the cell surface*. 2004. **165**: p. 735-746.
12. Goodwin, J., et al., *Ras diffusion is sensitive to plasma membrane viscosity*. Biophysical Journal, 2005. **89**: p. 1398-1410.
13. Shvartsman, D., et al., *Cyclodextrins but not compactin inhibit the lateral diffusion of membrane proteins independent of cholesterol*. Traffic, 2006. **7**(7): p. 917-926.
14. Pucadyil, T. and A. Chattopadhyay, *Cholesterol modulates ligand binding and G-protein coupling to serotonin 1A receptors from bovine hippocampus*. Biochim Biophys Acta, 2004. **1663**: p. 188-200.
15. Torigoe, C., et al., *The influence of actin microfilaments on signaling by the receptor with high-affinity for IgE*. Molecular Immunology, 2004. **41**: p. 817-829.

Chapter IV: Continuous fluorescence depletion anisotropy measurement of protein rotation¹

4.1. Overview

Protein rotation in viscous environments can be measured by fluorescence depletion anisotropy (FDA) which combines long lifetimes of chromophore triplet states with the sensitivity of fluorescence excitation and detection. FDA achieves sensitivity well beyond that attainable by the more common technique of time-resolved phosphorescence anisotropy (TPA). We have now combined benefits of both time-domain and frequency-domain FDA into a single continuous technique (CFDA). Intensity and polarization of a single laser beam are modulated continuously according to a complex, repeating waveform. Fluorescence signals excited from triplet-forming fluorescent probes are digitized over recurring waveform periods by a high-speed signal averager. CFDA experiments typically involve substantial ground state depletion. Thus signals, unlike those of TPA, are not linear in the exciting light intensity and simple data analysis based on such linearity is not appropriate. An exact solution of the coupled diffusion and triplet production/decay equation describing CFDA within individual data points has been combined with simulated annealing optimization to extract triplet and anisotropy decay kinetics from experimental data. Related calculations compare possible excitation waveforms with respect to rotational information provided per fluorescence photon. We present CFDA results for the model system of eosin conjugates of carbonic anhydrase, BSA and immunoglobulin G in 90% glycerol at various temperatures and initial cellular results on eosin-IgE bound to 2H3 cell Type I

¹ Zhang, D., et al., *Continuous Fluorescence Depletion Anisotropy Measurement of Protein Rotation*. Journal of Fluorescence, 2018. **28**: p. 533-542.

Fcε receptors. We explore how CFDA reflects rotational parameters of heterogeneous systems and discuss challenges of extending this method to single cell microscopic measurements.

4.2. Introduction

Changes in the motions of integral membrane proteins reflect and/or modulate primary events in cellular activation [1, 2]. In particular, rates of rotational diffusion [3] reflect protein interactions, aggregation and conformation and so are sensitive measures of the size and microenvironment of these molecules. Experimentally, unhindered rotational diffusion occurs on the microsecond timescale and has been studied using various methods, including linear dichroism, delayed fluorescence, time-resolved phosphorescence anisotropy (TPA) and fluorescence depletion anisotropy (FDA) [4-6]. However, cellular studies demand robust, broadly-applicable methods and only the latter two of these methods have been widely used.

The anisotropy function describes the orientational asymmetry of any molecular distribution and is in fact the simplest quantitative measure of the deviation of the emission dipole distribution from spherical symmetry. Emission anisotropy r is calculated as $(I_{\parallel} - I_{\perp}) / (I_{\parallel} + 2I_{\perp})$ where I_{\parallel} and I_{\perp} are luminescence intensities polarized parallel and perpendicular, respectively, to the polarization of an exciting pulse. The anisotropy of a freely-rotating *spherical* species in solution decays mono-exponentially with a rotational correlation time τ of $1/6D = \eta V_h / kT$ where D is the rotational diffusion constant, η is the viscosity and V_h is the molecular hydrated volume. Chromophores in the asymmetric environment of a membrane can exhibit multiple rotational correlation plus a non-decaying component or “limiting” anisotropy.

However, distinct rotational correlation times can rarely be resolved in practice, so it is most common to represent apparent decay as $r = r_{\infty} + (r_0 - r_{\infty}) \exp(-t/\phi)$ where ϕ is an average rotational correlation time. This quantity clearly reflects size, asymmetry, environment and interactions of the rotating molecule.

Protein rotation in viscous solutions or on cell surfaces can be measured by various methods. Time-resolved phosphorescence anisotropy (TPA) measurements of protein rotation are analogous to the better-known time-resolved fluorescence anisotropy (TFA) methods. However, singlet lifetimes of fluorescent probes are typically only a few nanoseconds, hence TFA is limited to measuring low-nanosecond timescale rotations. Apart from low quantum yields of suitable phosphors, the necessarily long triplet state lifetimes imply low photon fluxes, even at saturation, and long-wavelength phosphorescence is poorly detected by most high-speed detectors. By contrast, fluorophores typically exhibit high quantum yields, nanosecond lifetimes and mid-visible fluorescence emission.

It is thus natural to ask if the long lifetime of triplet states could be combined with the sensitivity of fluorescence excitation and detection. Such a technique, Fluorescence Depletion Anisotropy, was proposed by Peter Garland [7] and used subsequently by our group in a number of studies [8-10]. The method depends upon fluorophores like eosin isothiocyanate (EITC) which have substantial quantum yields both for triplet formation and for prompt fluorescence [11]. Rotationally-mobile macromolecules labeled with such a chromophore are first irradiated by a low-intensity, linearly-polarized probe beam (Figure 4.1 upper panel), *e.g.* from an Ar-ion laser at 514.5 nm. The resulting steady-state fluorescence is proportional to the number of

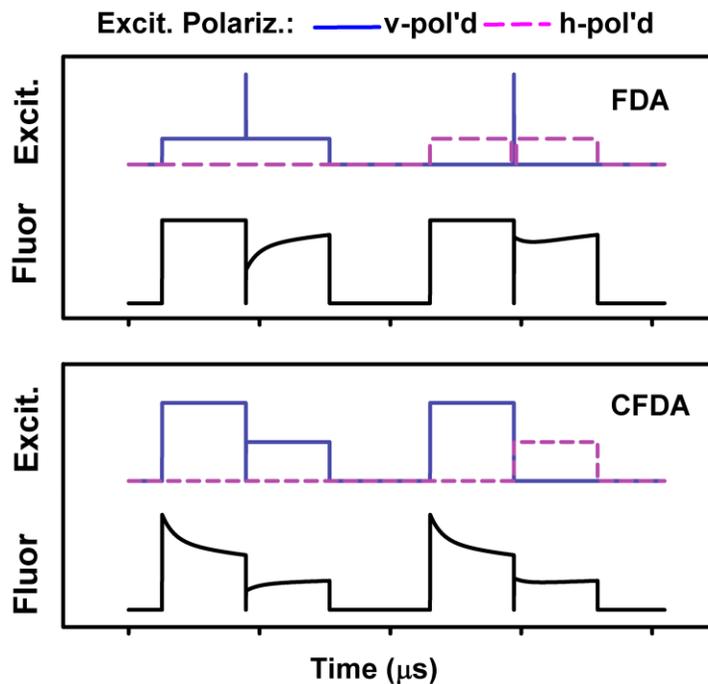


Figure 4.1: Rotation information in pulse fluorescence depletion anisotropy (FDA) and continuous fluorescence depletion anisotropy (CFDA) data. Blue (solid) lines and red (dashed) lines in upper panel halves marked “lasers” indicate vertical and horizontal polarizations, respectively.

ground state chromophores whose absorption transition dipoles are parallel to the probe polarization. The sample is then subjected to a brief pulse of high-intensity, linearly polarized light. A pulsed Nd:YAG laser producing mJ pulses at 532 nm is appropriate to examine 1mm² of sample. Under this intense illumination, a substantial fraction of chromophores undergo intersystem crossing to the triplet state. These triplets can exist for several hundred microseconds and, during this period, they cannot be excited to fluorescence by the probe beam. Thus, immediately after the pump pulse, there is asymmetric depletion in sample fluorescence which recovers back to the original steady state by the mechanisms of triplet decay and rotational reorientation. By recording traces in the presence and absence of both pump and probe beams, undesired signals from sample autofluorescence, gating transients, etc. are cancelled automatically cancelled.

The relative advantage of FDA over TPA can be estimated by considering the major photophysical processes in a three-level system (Figure 4.2). The maximum FDA fluorescence signal reflecting the non-fluorescent, slowly-decaying triplet state can be compared with the maximum phosphorescence signal arising from this or from another chosen chromophore. If one ignores reverse intersystem crossing k_R , the maximum fluorescence rate arising in FDA from triplet decay is approximately $\Phi_f \Phi_i / \tau_t$ where, for the FDA chromophore, Φ_f and Φ_i are the quantum yields for prompt fluorescence and inter-system crossing, respectively, and τ_t is the triplet lifetime. The maximum phosphorescence rate in TPA is approximately one phosphorescence quantum yield Φ_p of photons per phosphorescence lifetime τ_p or Φ_p / τ_p . The relative signal advantage of FDA over TPA is thus $(\Phi_f \Phi_i / \Phi_p) (\tau_p / \tau_t)$. Comparing eosin in FDA with erythrosin in TPA, this advantage is 10- to 100-fold. This sensitivity enhancement relative

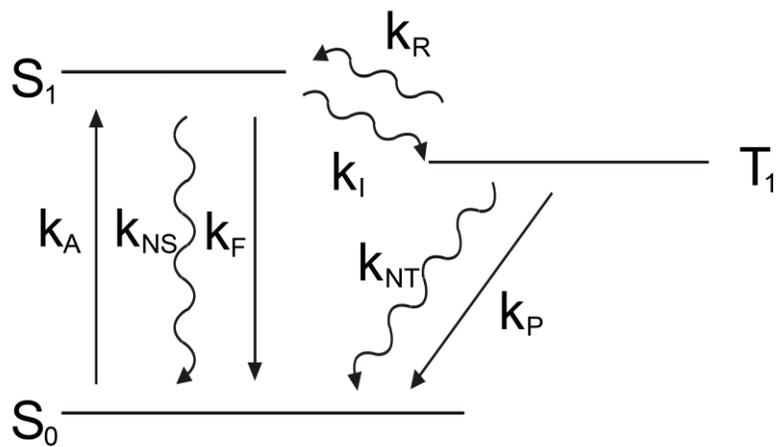


Fig. 4.2: Jablonski diagram showing major photophysical processes in a three-level system.

to TPA has allowed protein rotation measurements via FDA on individually-selected cells [12]. Nonetheless there are practical difficulties in time-domain FDA measurements. Apart from the photophysical limitations on signal intensities just described, there are instrumental constraints on data acquisition rate, detector saturation from excitation pulse and waste of fluorescence photons during excitation pulse.

While the above limitations might be approached by improved dyes, faster lasers combined with fast photon counting and more effective detector gating, respectively, frequency domain measurements may also appear to address at least the latter two issues. We have previously investigated frequency-domain FDA where a continuous-wave laser's intensity and polarization are modulated by frequencies ν and w , respectively [13]. Unfortunately, intrinsic non-linearity of FDA spreads rotation information into many frequencies, *e.g.* ν , w , 2ν , $2w$, $\nu-w$, $\nu+w$, 2ν , $2w$, $2(\nu-w)$, $2(\nu+w)$, $2\nu-w$, etc. Thus efficient collection of distributed rotational information actually amounts to performing a time-domain experiment.

One may wonder if advantages of time- and frequency-domain approaches could be combined into single technique. Intensity and polarization of a single laser beam would be modulated continuously according to a complex, repetitive waveform and fluorescence signals averaged over recurring waveform periods by a low rearm-time signal averager. This would offer prospective advantages of no gating, no wasted fluorescence, use of a single laser and data collection in a continuous experiment. We have previously realized such a technique, continuous fluorescence depletion anisotropy (CFDA), and described preliminary experiments measurements using the approach [14]. In the present paper we describe an improved approach

to analysis of resulting data and present rotational information obtained on glycerol solutions of well-known proteins at various temperatures as well as initial cellular results on eosin-IgE bound to 2H3 cell Type I Fcε receptors. Our goal in this project is to evaluate the suitability of CFDA for further development aimed at measuring protein rotation on cell surfaces.

4.3. Theory underlying data analysis

Consider an L-format optical system with excitation along the z-axis and emission acquired along the x-axis. Samples consist of freely-rotating macromolecules to which chromophores are rigidly attached. The absorption and fluorescence emission transition dipoles are assumed here to be collinear, a reasonable approximation for the xanthine dyes such as fluorescein and eosin. However, a similar treatment can be developed for non-collinear transition dipoles. On the timescale of experiments and at light intensities employed, chromophores can be assumed to exist only in the singlet ground state and the first triplet excited state.

The rate constant for triplet formation by light of unit intensity polarized parallel to the absorption transmission dipole is given by

$$k_b = \frac{3 \cdot 2303 \epsilon_M \Phi_T hc}{\lambda N_A} \quad (4.1)$$

where ϵ_M is the molar absorptivity, Φ_T is the triplet quantum yield, h is Planck's constant, c is the speed of light, N_A is Avogadro's number and λ is the wavelength. The factor of 3 arises since ϵ_M is defined for randomly-oriented molecules.

The basic equation for CFDA is expressed in terms of the distribution function $c(\theta, \phi, t)$ of ground state chromophore transition dipoles

$$k_r \nabla^2 c(\theta, \phi, t) + k_d (1 - c(\theta, \phi, t)) - k_b I(\theta, \phi, t) c(\theta, \phi, t) = \frac{\partial c(\theta, \phi, t)}{\partial t} \quad (4.2)$$

where k_r is the rotational diffusion constant of the chromophore, k_d is the decay rate of triplet chromophores to the ground-state k_b , is the rate of excitation of ground-state chromophores to the triplet state and $I(\theta, \phi, t)$ is the intensity of time- and polarization-varying exciting light on the sample.

Since $\nabla^2 Y_n^m(\theta, \phi) = -n(n+1)Y_n^m(\theta, \phi)$ we can express c as a sum of even-order spherical harmonics $Y_n^m(\theta, \phi)$

$$c(\theta, \phi, t) = \sum_{n=0}^{\infty} \sum_{m=-n}^{n} a_n^m(t) Y_n^m(\theta, \phi) \quad (4.3)$$

Without loss of generality, we can assume that $c(\theta, \phi, 0) = 1/\sqrt{4\pi}$ so that $a_0^0(0) = 1$. For a given set of k_d , k_r and k_b , combining Equations 4.2 and 4.3 yields

$$\frac{\partial a_n^m(t)}{\partial t} - k_d a_n^m(t) = -n(n+1)k_r a_n^m(t) - k_b I(\theta, \phi, t) a_n^m(t) \quad (4.4)$$

Sample illumination $I(\theta, \phi, t)$ affects Y_n^m in the RHS of Equation 4.3. Our illumination consists of mixtures of x- and y-polarized light of intensities $I_x(t)$ and $I_y(t)$, respectively, so that $I_x(t) = \sin^2 \theta \cos^2 \phi I_{x0}(t)$ and $I_y(t) = \sin^2 \theta \sin^2 \phi I_{y0}(t)$. Interaction of such light with a given Y_n^m produces up to 9 spherical harmonics ranging from Y_{n-2}^{m-2} up to Y_{n+2}^{m+2} . Explicit equations for the magnitudes of these new Y_n^m can be derived from the properties of spherical harmonics or their underlying associated Legendre polynomials [15]. The distribution function c can be obtained as a power series in $k_b I$. If n is the order of a given term in that series, then the term will be comprised of spherical harmonics up to order $2n$. Such a series converges unconditionally and we have explored this approach. However, convergence is slow when k_d and/or k_r are large and an alternate solution is desirable.

The indices n and m of a spherical harmonic Y_n^m uniquely determine its position i in a list where all Y_n^m are ordered, first, by increasing n and, then, by increasing m , as shown. Hence all spherical harmonics can be described as members of a one-dimensional array. Simple relations allow i to be evaluated from n and m and n and m from i and we can consider c to be a function of a fixed number of even spherical harmonics up to some order n_{\max} . Then the distribution function c will consist of $(n_{\max}/2 + 1)^2$ terms. As a concrete example, if n_{\max} is 2, there are four terms. For simplification, let $I_{s0}(t) = I_{x0}(t) + I_{y0}(t)$ and $I_{d0}(t) = I_{x0}(t) - I_{y0}(t)$.

$$\begin{vmatrix}
\frac{1}{3}k_b I_{s0}(t) & \frac{1}{\sqrt{30}}k_b I_{d0}(t) & \frac{-1}{3\sqrt{5}}k_b I_{s0}(t) & \frac{1}{\sqrt{30}}k_b I_{d0}(t) \\
\frac{1}{\sqrt{30}}k_b I_{s0}(t) & \frac{3}{7}k_b I_{d0}(t) + k_d + 6k_r & \frac{-1}{7}\sqrt{\frac{2}{3}}k_b I_{s0}(t) & 0 \\
\frac{-1}{3\sqrt{5}}k_b I_{s0}(t) & \frac{-1}{7}\sqrt{\frac{2}{3}}k_b I_{d0}(t) & \frac{5}{21}k_b I_{s0}(t) + k_d + 6k_r & \frac{-1}{7}\sqrt{\frac{2}{3}}k_b I_{d0}(t) \\
\frac{1}{\sqrt{30}}k_b I_{s0}(t) & 0 & \frac{-1}{7}\sqrt{\frac{2}{3}}k_b I_{s0}(t) & \frac{3}{7}k_b I_{d0}(t) + k_d + 6k_r
\end{vmatrix}
\begin{vmatrix}
a_0^0(t) \\
a_2^{-2}(t) \\
a_2^0(t) \\
a_2^2(t)
\end{vmatrix}
-
\begin{vmatrix}
a_0^0(0) \\
a_2^{-2}(0) \\
a_2^0(0) \\
a_2^2(0)
\end{vmatrix}
=
\begin{vmatrix}
k_d \\
0 \\
0 \\
0
\end{vmatrix}
\tag{4.5}$$

or

$$\mathbf{B}(t) \mathbf{A}(t) - \mathbf{A}(0) = [k_d, 0, 0, \dots]^T \equiv \mathbf{f}(t)$$

Thus an inhomogeneous equation of the form shown can be used to describe the chromophore distribution function to whatever precision desired.

Solution of Equation 4.5 at a given point follows standard procedures [16]. For the matrix \mathbf{B} comprising the LHS of Equation 4.5, eigenvalues λ_i are calculated as a vector of length n and corresponding eigenvectors as an $n \times n$ matrix $\mathbf{\Lambda}$, respectively. An n -vector $\boldsymbol{\lambda}(t)$ whose elements λ_i are $\exp(\lambda_i t)$ is generated and a fundamental matrix $\mathbf{M}(t)$ is constructed as $\mathbf{M}(t) = \mathbf{\Lambda} \mathbf{Diag}[\boldsymbol{\lambda}(t)]$. The solution $\mathbf{A}(t)$ of the inhomogeneous equation at time t following an initial value $\mathbf{A}(t_0)$ is then given by

$$\mathbf{A}(t) = \mathbf{M}(t) \mathbf{M}^{-1}(t_0) \mathbf{A}(t_0) + \mathbf{M}(t) \int_{t_0}^t \mathbf{M}^{-1}(s) \mathbf{A}(s) \mathbf{f}(s) ds \tag{4.6}$$

As written, this system of inhomogeneous linear first-order differential equations has non-constant coefficients $I_{x0}(t)$, $I_{y0}(t)$ and, as such, is only soluble with substantial difficulty. A satisfactory alternate approach is to consider the equation pointwise. Since the exciting laser intensity remains constant during each measurement point, the equation may readily be solved at each point using as initial conditions the results of the previous point. The entire experimental

signal can thus be generated for specified rate constants and other parameters as desired. The time-dependent orientational distribution function $c(\theta, \varphi, t)$ can thus be evaluated as

$$c(\theta, \varphi, t) = \mathbf{A}(t) \cdot |Y_0^0(\theta, \varphi), Y_2^{-2}(\theta, \varphi), Y_2^0(\theta, \varphi), Y_2^2(\theta, \varphi), \dots|^T \quad (4.7)$$

to any desired precision by incorporating spherical harmonics up to a sufficiently large order n .

The last computational issue is the fluorescence elicited from the sample by the illumination. Both x- and y-polarized light of unit intensity can be decomposed into four spherical harmonics.

$$\begin{aligned} x^2 &= \sqrt{4\pi} [Y_0^0 - \sqrt{\frac{3}{10}} Y_2^0 + \sqrt{\frac{1}{5}} (Y_2^2 + Y_2^{-2})]; \\ y^2 &= \sqrt{4\pi} [Y_0^0 - \sqrt{\frac{3}{10}} Y_2^0 - \sqrt{\frac{1}{5}} (Y_2^2 + Y_2^{-2})] \end{aligned} \quad (4.8)$$

Fluorescence elicited by light of either polarization is calculated using the orthonormality of spherical harmonics. The choice for leading constants is such that initial fluorescence from unit excitation intensity is 1. Our experimental configuration uses a magic-angle polarizer to measure total emitted fluorescence [17] so that, if E is the efficiency of fluorescence collection, the measured total fluorescence F is

$$\begin{aligned} F = E & \left[I_{x0}(t) + I_{y0}(t) \right] a_0^0(t) - \sqrt{\frac{1}{5}} \left[I_{x0}(t) + I_{y0}(t) \right] a_2^0(t) \\ & + \sqrt{\frac{1}{5}} \left[I_{x0}(t) - I_{y0}(t) \right] \left[a_2^2(t) + a_2^{-2}(t) \right] \end{aligned} \quad (4.9)$$

We assume that triplet decay and molecular rotation are independent of each other. Then triplet decay is characterized by n_d multiple decay rates k_{di} , with fractions f_{dj} and molecular rotation is specified by n_r multiple rotational decay rates or correlation times k_{rj} with fractions f_{ij} and the overall signal $F_o(t)$ becomes

$$F_o(t) = \sum_{i=1}^{n_d} \sum_{j=1}^{n_r} f_i f_j F(k_{di}, k_{rj}, t) \quad (4.10)$$

The rotational rate constants k_{rj} are rotational diffusion constants. We assume only a single rotational diffusion constant, i.e. $k_{ri} = k_r$ for all i .

Data analysis according to Equation 4.10 involves considerable computation, the amount of which increases strongly with the number of constants fitted. We therefore fitted CFDA data to seven fixed lifetimes, typically 8, 24, 80, 240, 800, 2400 and 8000 μ s, evaluating the best-fitting decay fraction of each lifetime. . The rotational behavior of the macromolecule is modeled assuming a limiting anisotropy r_∞ representing rotationally-immobile species and amplitude $(r_0 - r_\infty)$ decaying as a single exponential, *i.e.* with a single rotational correlation time. We can evaluate the rotation fractional amplitudes f_{ij} for 3 rotation rates, the first k_{r1} being arbitrarily fast and representing sub-microsecond motions, the second k_{r2} representing overall protein rotation and the third k_{r3} being zero and so representing the rotationally-immobile fraction of chromophores. To maintain consistency with TPA and TFA experiments, we can then report $(f_{r2} + f_{r3})$ as the “initial anisotropy” r_0 , $1/6k_{r2}$ as the protein RCT and f_{r3} as the “limiting anisotropy” r_∞ .

Even if the decay rates k_{di} are considered fixed, the preceding treatment nonetheless specifies a large number of independent parameters which must be adjusted to optimize agreement between the observed and calculated fluorescence traces. We initially employed the Marquardt non-linear fitting procedure [18] but found that searches became trapped in local minima. Even when small random increments were added to parameters to restart trapped searches, global minima were not reached.

Satisfactory optimizations were achieved using the so-called Simulated Annealing method of Goffe *et al.* [19]. This procedure simulates the process by which melted substances, through slow cooling, escape from defect traps to reach a minimum-energy crystalline state. Briefly, a system is assigned a starting “temperature” and initial parameter set. A random variation in parameters is introduced and a system “energy”, here the average square deviation between observed and calculated fluorescence, is evaluated. If the parameter change reduces the energy, *i.e.* improves the fit, then that parameter change is accepted and a new parameter change attempted. However, if the energy is increased by the parameter change, *i.e.* $\Delta E > 0$, then that change may be rejected *or* accepted. The quantity $\exp(-\Delta E/aT)$ where a is a constant is compared to a random number between 0 and 1 and the parameter change accepted if the random number is smaller. Thus, parameter changes that only slightly increase energy are most likely to be accepted, but changes substantially increasing energy nonetheless have finite probabilities of acceptance. As the search proceeds, the temperature and the parameter step sizes are reduced to simulate the annealing process.

4.4. Materials and methods

4.4.1. Apparatus and data acquisition

The apparatus used in CFDA experiments is shown in Figure 4.3 and has been described in a preliminary report [14]. Continuous wave excitation at 514.5 nm is provided by the vertically-polarized TEM₀₀ output from a Coherent Radiation Innova 90 argon ion laser. A Lasermetrics 3031 transverse Pockels cell (Lasermetrics Inc./FastPulse Technology, Teaneck, NJ), driven by a Lasermetrics AF-3 driver, rotates the beam polarization in response to a waveform generated in timing hardware, the response time being less than 200ns for a 90° rotation. The beam intensity is then adjusted by a Coherent 304 acoustic-optic modulator (AOM; Coherent Inc., Modulator Division, Danbury, CT) in response to an input waveform. Maximum modulated power at the sample is approximately 100 mW. To increase laser intensity at the sample, a 2x Galilean telescope reduces the beam $1/e^2$ diameter to approximately 0.7 mm. A Tektronix AF320 Arbitrary Function Generator generates waveforms for the Pockels cell and AOM. A half-wave plate set at 22° adjusts the polarization of the un-rotated and rotated beams to +45° and -45° with respect to the vertical. Samples are examined in 5x5 mm Suprasil cuvettes (Helma). Sample fluorescence is collected at 90° to the excitation axis and through a polarizer set at 35° to the vertical. Fluorescence of any polarization is thus collected with equal efficiency [17] so that the measured signal is proportional to the total emitted fluorescence. Depletion data thus reflect *only* the orientation of *absorption* transition dipoles and, as such, are true absorption anisotropies. Scattered light and phosphorescence are removed by a K₂Cr₂O₇ chemical filter, a Schott KV550 filter and a 600 nm short-pass interference filter and fluorescence is detected

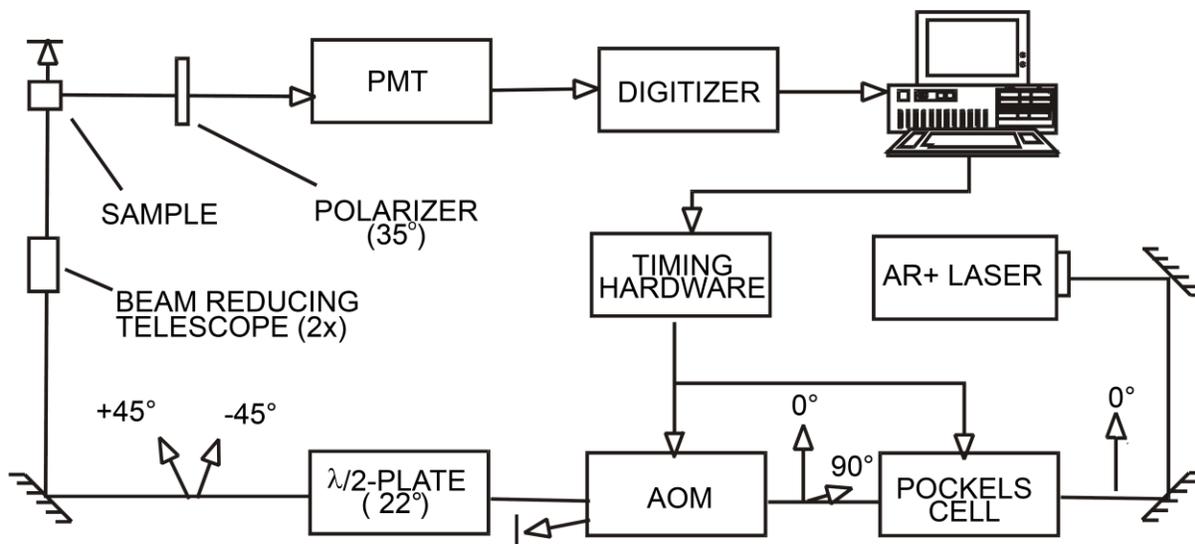


Figure 4.3: Apparatus for Continuous Fluorescence Depletion Anisotropy (CFDA) measurements on bulk samples. The functions of the various components are described in the text.

using an EMI9816 photomultiplier tube (PMT; Thorn EMI Gencom Inc., Plainview, NY). PMT signals are amplified by a Tektronix 473 oscilloscope and averaged by an EG&G 9826 signal averager with 0.6 μ s re-arm time. In a typical experiment 256k points, 256 ns/pt, are digitized per trace and 4096 traces are averaged at a repetition rate of 16 traces/sec.

Test systems exhibiting rotational correlation times comparable to many membrane proteins were prepared from 200-500 nM solutions of eosin isothiocyanate-conjugated proteins [20] in approximately 90% glycerol and examined over 4-37°C, with actual glycerol concentrations in samples determined by measurement of refractive index. Bovine IgG, bovine serum albumin (BSA) and bovine carbonic anhydrase (CA) were obtained from Sigma Chemical Co. (St. Louis, MO).

Independent measurements of eosin triplet, *i.e.* phosphorescence, lifetime were performed using an IBH 5000U fluorescence lifetime spectrometer equipped with xenon flash lamp and multichannel scaling for phosphorescence lifetime measurements. Excitation was at 526 nm and emission was recorded at 680 nm. Band widths were 16 nm and 32 nm, respectively. Data from ~110,000 lamp flashes were acquired in 1000 0.5 μ s channels and were analyzed beginning at 5 μ s after the lamp flash to avoid including the large fluorescence transient immediately after each flash. Decay was fitted to a single exponential decay model.

4.4.2. Waveform Selection

Any light waveform where intensity changes both with and without polarization change can yield rotation information and many such optical waveforms are possible. Waveform optimization must be with respect to a specific parameter for a typical sample. A suitable figure of merit for optimization is $(s^2\Sigma F)^{-1}$, the reciprocal of the squared standard error for a particular parameter multiplied by the total intensity in the detected waveform from which the parameter is estimated. This quantity increases as the desired parameter is better defined from a constant amount of sample fluorescence.

We calculated this figure of merit for accuracy of determining rotational correlation time for the waveforms illustrated in Figure 4.4. The upper waveform (a) where the polarization never changes contains no rotational information. The next two waveforms (b, c) where four shorter sections in the latter replace two longer ones but with equal total power, exhibit equal figures of merit. Waveform (d) with four equal intensity segments, one of which involves a polarization change, affords substantially, more information. The chirp waveform (e) is more efficient, where intensity changes over the range of triplet decay times both with and without polarization change over the range of rotational correlation times. The most efficient waveform is (f). Acousto-optic modulators exhibit substantial nonlinearity, so it is difficult to determine the exact optical intensity provided by a given electrical waveform. However, the uniform-intensity waveform (f) provides even more information than the chirp, while avoiding such calibration issues. Hence we currently use a single illuminated intensity to avoid issues of AOM non-linearity.

4.4.3. Data obtained

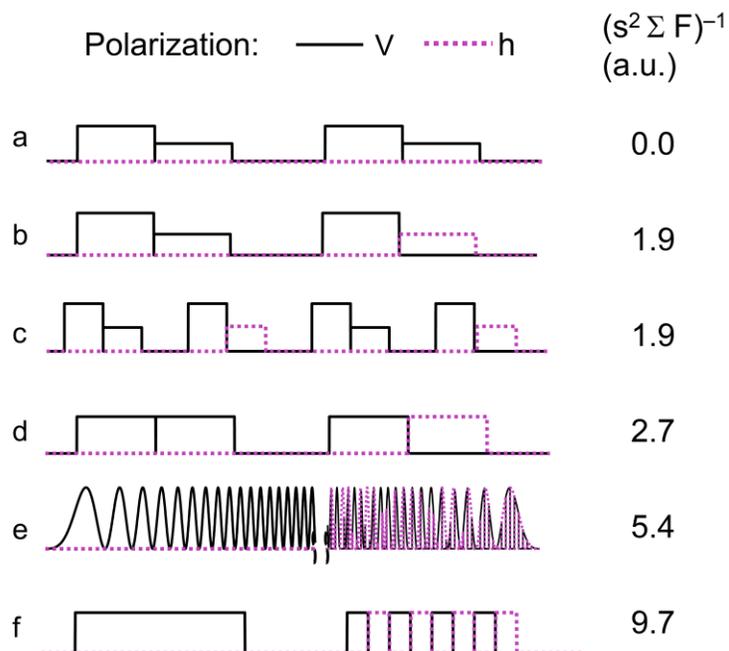


Figure 4.4: Relative efficiencies of various waveform types in using available fluorescence photons in evaluating the rotational correlation time of a rotating species in solution. All traces represent 4096 points with triplet decay times of 3, 30 & 300 points and fractional amplitudes of 0.3, 0.4 & 0.3 respectively. Rotational correlation time is 30 points in all cases.

Figure 4.5 shows a complete example waveform used to generate rotation data. The illuminated portions of the waveform were separated by substantial dark intervals. This was done to guarantee complete decay of light-induced triplet states between illuminated sections, thus possibly simplifying analysis. However, we emphasize that this is not necessary as our analysis scheme applies to *any* repetitive waveform. Hence, shortening the dark intervals shown in Figure 4.5 would increase the rate of data acquisition by several-fold.

Rotation information is available whenever the polarization of the exciting light is changed. This phenomenon is most clearly in Figure 4.6 where the sum of Figure 4.5 sections b and d is subtracted from the sum of sections a and c. This removes the large fluorescence decrease over time resulting from the exciting light conversion of chromophores to triplets. The actual dependence of fluorescence signal on the intensity and polarization of the exciting light is exceptionally complex as shown in Section 2 and inclusion of this complexity is essential in quantitative analysis of CFDA data.

Data analysis involves optimizing agreement between full 4-section fluorescence trace such as shown in Figure 4.5 and those calculated Eq. 10. The standard deviation of experimental points about best-fitted curves for full-scale fluorescence signals of over 1×10^6 is typically 500-1000, a relative standard error less than 0.1%. The quality of such agreement can be visualized by examination of Figure 4.6.

4.5. Results and discussion

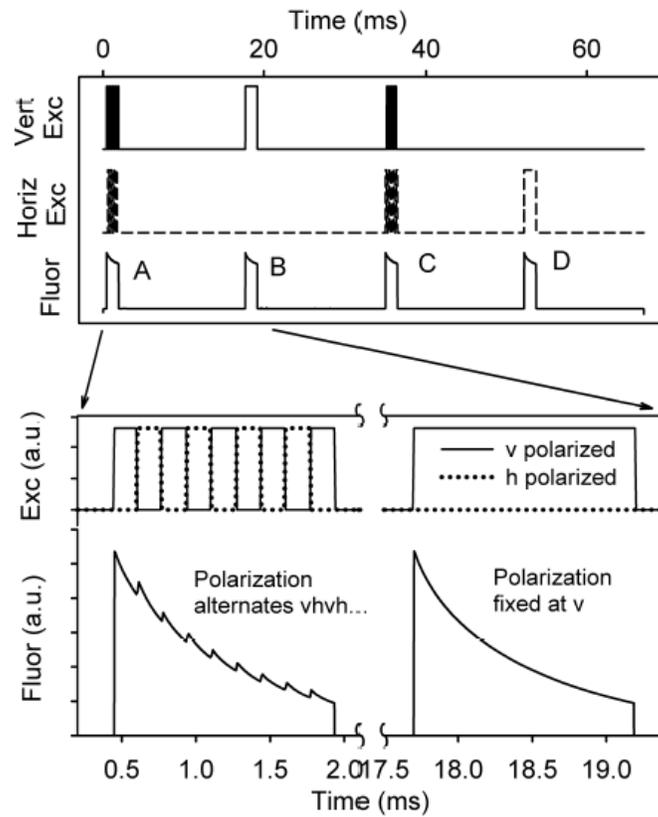


Figure 4.5: Excitation waveform and raw fluorescence data for CFDA measurement of EITC-BSA rotation at 4°C in 90% glycerol.

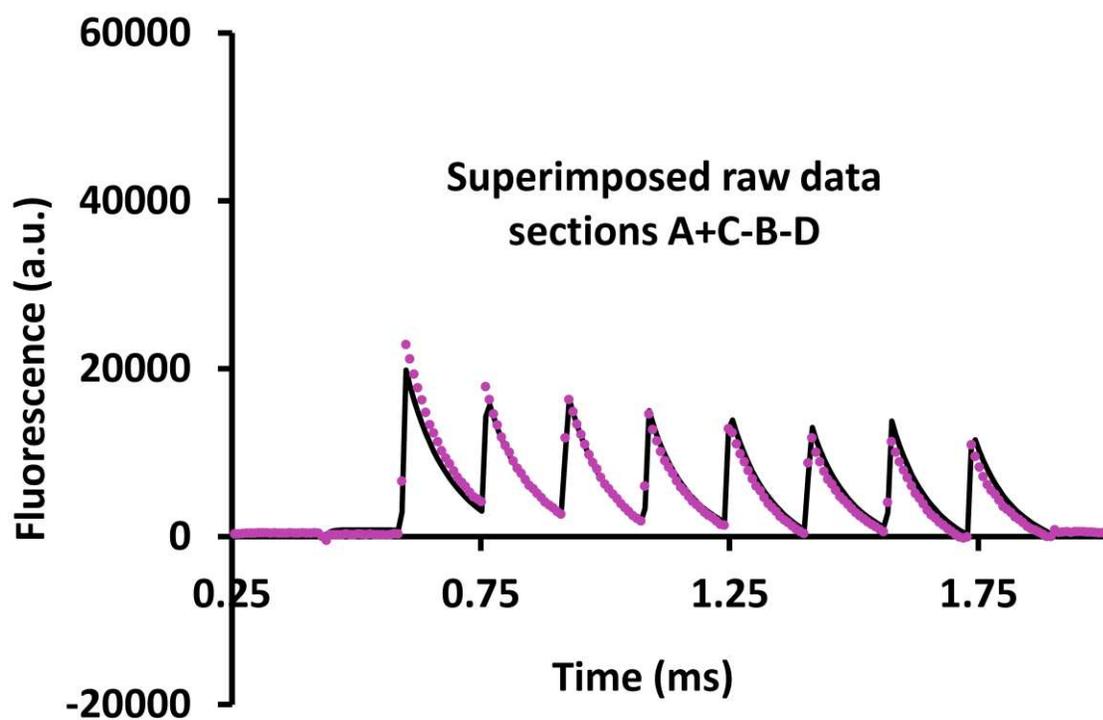


Figure 4.6: Excitation waveform and raw fluorescence data for CFDA measurement of EITC-BSA rotation at 4°C in 90% glycerol. Dotted lines indicate raw data while the smooth line represents fitted curves calculated by Equation 4.10.

4.5.1. Protein rotation versus temperature and viscosity

We examined by CFDA the rotation of three proteins, IgG, BSA and CA, in ~90% glycerol over 4-37 °C. Table 4.1 shows the rotational correlation times τ of eosin-conjugates of these proteins. Because neither the hydrated partial specific volume \bar{V}_h nor the effective axial ratio p of the proteins under these conditions is known, the expected values of the rotational correlation time cannot be predicted precisely. What can be said is that the rotational correlation time τ of a freely-rotating protein is given by

$$\tau = f\bar{V}_h \eta M / RT = f(\bar{V}_p + \delta\bar{V}_w) \eta M / RT \quad (4.11)$$

where \bar{V}_h , \bar{V}_p and \bar{V}_w are, the partial specific volumes of the hydrated protein, the protein alone, and its bound water, respectively, δ is the amount of bound water and f is a factor reflecting asymmetry of the hydrated particle [21]. Hence, a plot of τ vs. $\eta M/RT$ should have a slope of $f\bar{V}_h$ which should be approximately constant for a given protein in solutions of fixed composition. In fact, Figure 4.7 shows the RCT of each protein to vary linearly with $\eta M/RT$ and the slopes of such plots are indicated beside the best-fit lines and range from 1.9 to 2.9. Moreover, when the RCTs and corresponding $\eta M/RT$ for the three proteins are plotted on log-log plots, the slopes are 0.9-1.0, as predicted by Equation 4.11. While the uncertainties in RCT values are clearly substantial, increasing the waveform duty cycle and/or increasing the laser intensity at the sample should allow very substantial improvement.

Table 4.1: CFDA measurements of protein rotation in 90% glycerol solutions.^a Indicated uncertainties are the estimated standard deviations of the averaged RCTs.^b Uncertainties in these quantities are discussed in the text.

Sample	Temp	RCT (μs) ^a	r_0 ^b	r_∞ ^b	$t_{1/2}$ (CFDA, μs)	$t_{1/2}$ (from TPA lifetime, μs)
IgG	4	146.4 \pm 12.6	0.096	0.000	160	-
IgG	15	46.8 \pm 8.2	0.224	0.009	167	-
IgG	25	16.3 \pm 7.9	0.235	0.006	103	-
IgG	37	7.8 \pm 2.3	0.324	0.013	23	-
Bsa	4	101.3 \pm 11.4	0.238	0.000	280	177
Bsa	15	38.1 \pm 1.7	0.282	0.001	169	157
Bsa	25	20.4 \pm 2.0	0.267	0.004	171	142
Bsa	37	8.4 \pm 2.3	0.313	0.012	99	122
Ca	4	29.6 \pm 8.2	0.190	0.004	140	-
Ca	15	13.8 \pm 1.4	0.163	0.003	114	-
Ca	25	10.6 \pm 3.9	0.259	0.003	186	-
Ca	37	6.9 \pm 4.1	0.220	0.006	150	-

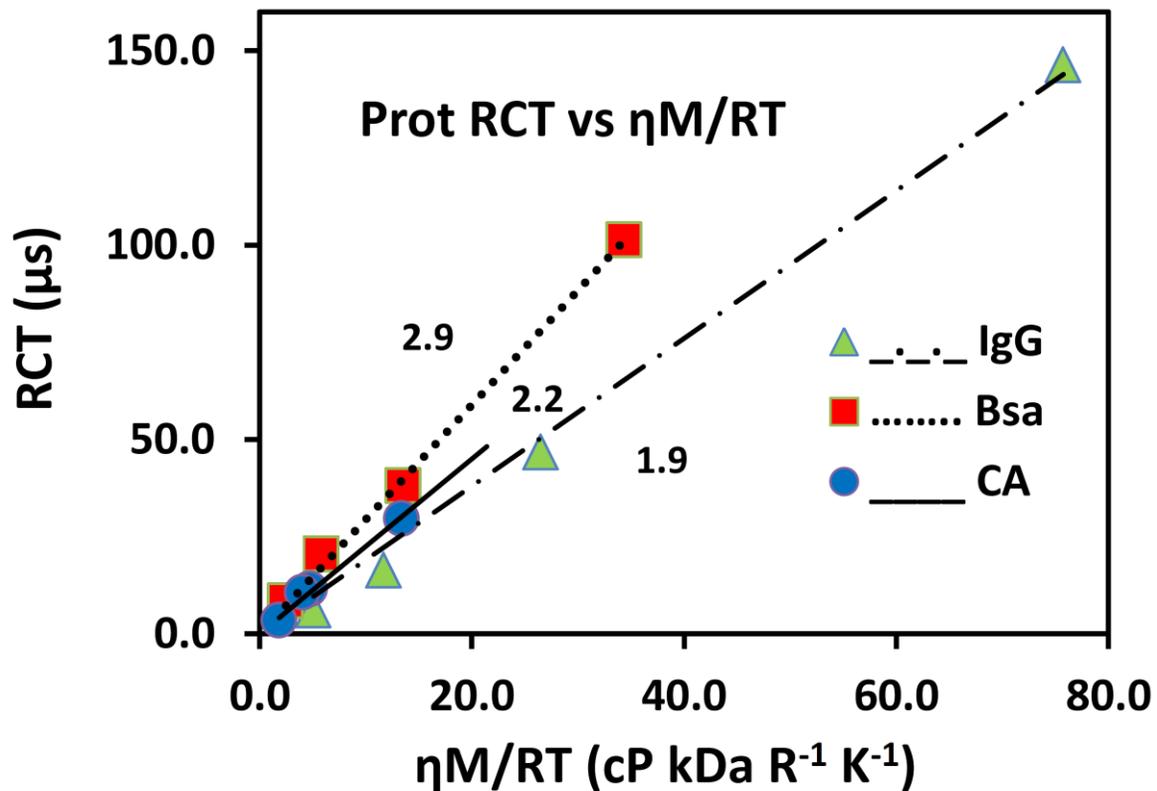


Figure 4.7: RCT of IgG, BSA and carbonic anhydrase in ~90% glycerol as functions of solution temperature and viscosity. Estimated uncertainties for points are given in Table 1. Slopes of the fitted lines are shown beside each plot.

Protein rotation in water has been examined by other investigators and their measured RCTs can be compared with our results. For carbonic anhydrase at room temperature, values of 14.6 ns [22] and 11.2 ns [23] have been reported. Immunoglobulin decay is strongly multi-exponential on account of the segmental flexibility of the Fab domains but, for the slow component reflecting overall molecular rotation, the mean of several studies [23-26] is approximately 130 ns. Several investigators have examined BSA, but a particularly careful examination by Ferrer *et al.* [27] using several methods gives a mean value of 40 ± 2 ns at 20 °C (1.00 cP).

Information concerning rotation of proteins in glycerol solutions is more limited. Using NMR methods Korchuganov [28] finds that addition of 20% glycerol almost doubles the RCT of barnase at 31.5 °C from 5.51 ns (0.77cP) to 9.38 ns (1.79 cP). For BSA in 95% glycerol at 23°C Yao *et al.* [29] obtain a value of 22 μ s while, for BSA in 92% glycerol at 6°C, Ferrer *et al.* [27] report 47 μ s. The RCTs of barnase and BSA both increase linearly with η/T as expected. In general, Priev *et al.* [30] suggest that glycerol causes compaction of the protein core but increases the size of the hydration layer.

Typical values for \bar{V}_p and δ in water are $0.73 \text{ cm}^3\text{g}^{-1}$ and 0.40 g H₂O per g protein [21], so that \bar{V}_h might be estimated at $1.13 \text{ cm}^3\text{g}^{-1}$. If proteins can be modeled satisfactorily as rotation ellipsoids, effects of asymmetry can be predicted from protein axial ratios p calculated from ultracentrifuge data or intrinsic viscosity measurements. For fluorescence anisotropy, Perrin's treatment [31] predicts three rotational correlation times for ellipsoids of revolution. In practice, for axial ratios of about 5 or less, the actual decay curve cannot be distinguished from a

single exponential. Axial ratios reported for IgG, BSA and CA in water at room temperature are 5.4 [32], 3.5 [33] and 5.4 [34], respectively. Thus, for an axial ratio of 3.5, relative to an equal-volume sphere, the RCTs and (fractional amplitudes) are predicted to be 0.97 (0.4), 1.92 (0.4) and 2.84 (0.2). Such a decay can be characterized by the harmonic mean of the rotational correlation times [31] or as the single exponential which can be best-fitted to the theoretical three-exponential decay. On the latter basis, for axial ratios of 3.5 and 5.4, single RCTs of 1.73 and 2.44, respectively, times those of equivalent spheres would be predicted. These quantities can be multiplied by \bar{V}_p to predict slopes of plots of RCT vs $\eta M/RT$ for the various proteins. For IgG, BSA and CA, predicted slopes are 2.8, 2.0 and 2.8 while experimental slopes, shown on Figure 4.7, are 1.9, 2.9 and 2.2, respectively. Taken together, these estimates of \bar{V}_h and f can largely explain the observation that RCTs of globular proteins are typically 2-3 times theoretical values [35] and that the slopes of plots of RCT vs $\eta M/RT$ in Figure 4.7 fall in this range.

4.5.2. Initial and limiting anisotropy values

Samples examined by CFDA exhibit initial anisotropies averaging 0.23 ± 0.06 while limiting anisotropies are 0.004 ± 0.007 . The first quantity is a typical absorption anisotropy for a chromophore conjugated to a protein [36], since nanosecond- timescale flexibility substantially reduces the initial anisotropy from its theoretical value of $2/5$. Limiting anisotropies for solutions of homogeneous macromolecules should be zero, so the standard deviation of measured anisotropies perhaps indicates the uncertainty in absolute anisotropies measured by these methods.

4.5.3. Lifetime values and distributions

To accommodate the multi-exponential triplet decay observed in CFDA, we have fitted this decay to a set of seven fixed lifetimes, typically 8, 24, 80, 240, 800, 2400 and 8000 μs . Examples of resulting decay distributions for EITC-BSA at 4°C and 37 °C are shown in Figure 4.8. At 4°C, the peak decay amplitudes are at 240 and 800 μs while, at 37°C, peak decays are at 80 and 240 μs . How such distributions relate to that lifetime(s) observed in a technique like time-resolved phosphorescence anisotropy (TPA) is complex. This is because, in TPA, an initial distribution of triplet chromophores is produced by a high-intensity and asymmetric pulse of nominally negligible width and this distribution decays by only by triplet decay and rotational randomization. In CFDA, the evolution of fluorescence over time also involves continuous pumping of molecules into the triplet state throughout the experiment by exciting light. This situation is properly modeled by Equation 4.10 but precisely how the resulting lifetime distributions such as those shown in Figure 4.8 relate to what might be observed in a TPA measurement is not clear. One possible comparison is to use the complete CFDA lifetime distributions to calculate apparent decay half-times $t_{1/2}$ and compare these quantities with 0.69 times the triplet lifetimes measured directly by TPA. Half-times ranging from approximately 100-300 μsec as evaluated by these two methods are shown in Table 4.1 and are in reasonable agreement, given the differences in the methods being compared.

4.5.4. Preliminary cellular results

The apparatus described in this paper was intended for measurements on cuvet-sized samples of

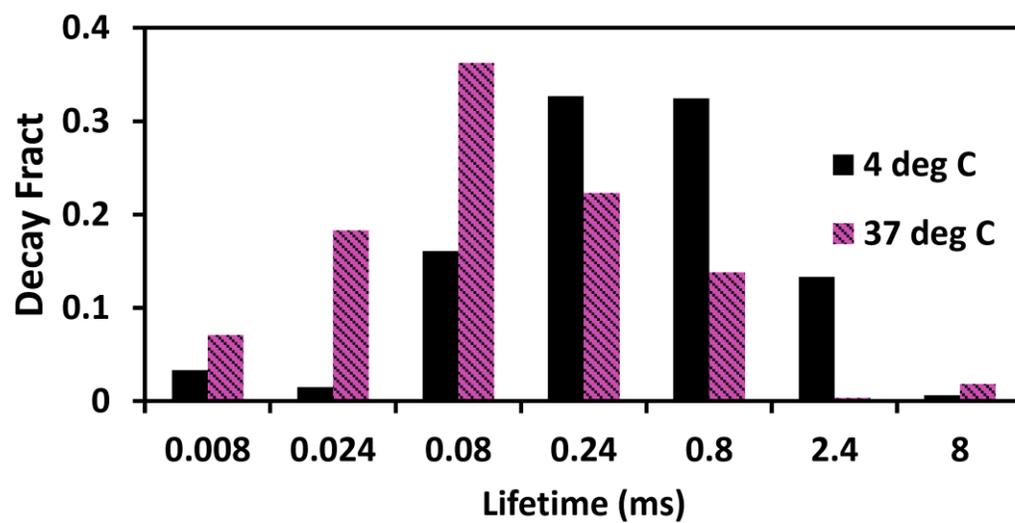


Figure 4.8: Triplet decay distributions for EITC-BSA in 90% glycerol at 4° and 37 °C.

protein solutions. However, suspended 2H3 cells labeled with EITC-A2-IgE were also examined at 4°C using single-intensity waveforms, albeit with small fluorescence signals and high noise (Figure 4.9). The fitted RCT is approximately 76 μs and this can be compared with a value of $82 \pm 17 \mu\text{s}$ obtained for erythrosin isothiocyanate-labeled A2-IgE in extensive TPA studies [37]. The CFDA method should be much more promising if implemented in a microscope-based system. The key factor in such expected improvement would be the much smaller illuminated area and resulting more intense illumination at the sample. For example, cuvet samples are examined in a Gaussian beam of diameter 0.7 mm. In a microscope implementation intended for examination of individual cells, the illuminated diameter might be 20 μm . For a given laser intensity, the peak intensity on a microscope sample would be about 800-fold greater than on a cell in the cuvet and hence a 20-50mW frequency-doubled Nd:YAG laser would provide almost ideal illumination.

4.6. Conclusions

Fluorescence Depletion Anisotropy (FDA) combines the long lifetime of triplet states with the sensitivity of fluorescence excitation and detection to measuring rotation of proteins in solution or on cell surfaces. Combination of time- and frequency-domain FDA in a single technique, Continuous Fluorescence Depletion Anisotropy (CFDA), provides protein rotation measurements via a continuous, single-laser method with no gating and no wasted photons. Rotational correlation times measured for common proteins in glycerol solutions exhibit the expected dependence on solution viscosity and temperature. Moreover, the technique appears to

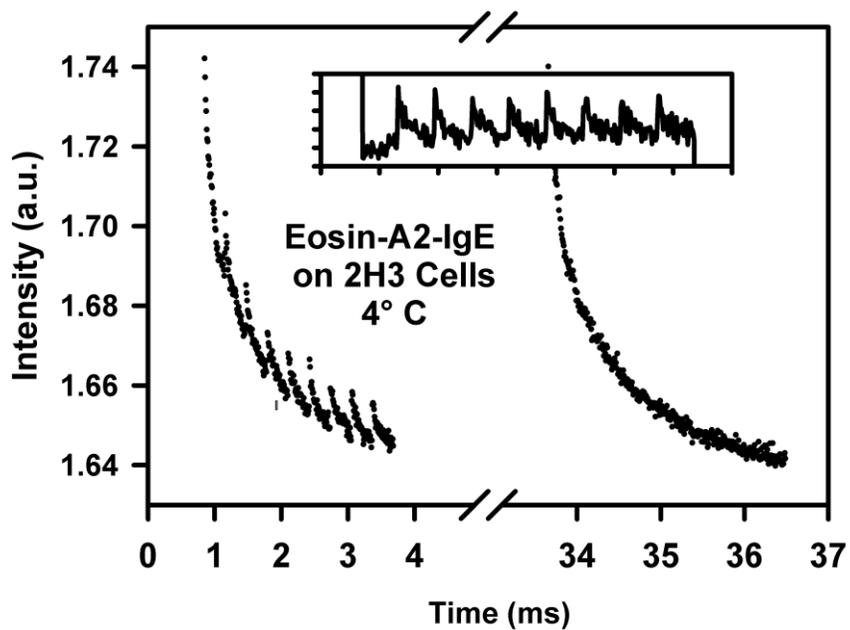


Figure 4.9: CFDA examination of suspended 2H3 mucosal mast cells at 4°C binding eosin isothiocyanate-conjugated A2 IgE. The fitted RCT is 76 μ s. The inset shows the difference between fluorescence excited by light of alternating polarization and that from fixed polarization. See Fig. 6.

have unique potential for measuring rotation of specific proteins on individual living cells, as will be examined in future studies.

4.7. References

1. Axelrod, D., *Lateral motion of membrane proteins and biological functions*. Journal of Membrane Biology, 1983. **75**: p. 1-10.
2. Edidin, M., *Rotational and translational diffusion in membranes*. Annual Review of Biophysics and Bioengineering, 1974. **3**: p. 179-201.
3. Hoffmann, W., M.G. Sarzala, and D. Chapman, *Rotational motion and evidence for oligomeric structures of sarcoplasmic reticulum Ca-activated ATPase*. Proceedings of the National Academy of Science (USA), 1979. **76**(8): p. 3860-3864.
4. Cherry, R.J., *Measurement of protein rotational diffusion in membranes by flash photolysis*. Methods in Enzymology, 1978. **54**: p. 47-61.
5. Cherry, R., *Rotational and lateral diffusion of membrane proteins*. Biochimica et Biophysica Acta, 1979. **559**: p. 289-327.
6. Greinhert, R., et al., *E-Type Delayed Fluorescence Depletion, a Technique to Probe Rotational Correlation Time in the Microsecond Range*. Journal Biochemical and Biophysical Methods, 1979. **1**: p. 77-83.
7. Johnson, P. and P.B. Garland, *Depolarization of fluorescence depletion. A microscopic method for measuring rotational diffusion of membrane proteins on the surface of a single cell*. FEBS Lett, 1981. **132**(2): p. 252-6.
8. Barisas, B.G., et al., *Rotation of plasma membrane proteins measured by polarized fluorescence depletion*. The International Society for Optical Engineering, Proceedings, 1990. **1204**: p. 765-774.

9. Yoshida, T.M., F. Zarrin, and B.G. Barisas, *Measurement of protein rotational motion using frequency domain polarized fluorescence depletion*. *Biophysical Journal*, 1988. **54**: p. 277-288.
10. Yoshida, T.M. and B.G. Barisas, *Protein rotational motion in solution measured by polarized fluorescence depletion*. *Biophysical Journal*, 1986. **50**(1): p. 41-53.
11. Garland, P.B. and C.H. Moore, *Phosphorescence of protein-bound eosin and erythrosin. A possible probe for measurements of slow rotational mobility*. *Biochemistry Journal*, 1979. **183**: p. 561-572.
12. Barisas, B.G., et al., *Rotational dynamics of Fc_ϵ receptors on individual 2H3 RBL cells studied by polarized fluorescence depletion*. *Biophysical Journal*, 1990. **75**: p. 671.
13. Yoshida, T.M., *Measurement of protein rotational diffusion using time-and frequency-domain polarized fluorescence depletion*. 1989, Colorado State University. p. 225.
14. Barisas, B.G. and H. Zhang, *Continuous fluorescence depletion anisotropy (CFDA) measurement of protein rotation*. *SPIE Proceedings*, 2001. **4260**: p. 140-148.
15. Jackson, J.D., *Mathematics for quantum mechanics: an introductory survey of operators, eigenvalues, and linear vector spaces*. 1st ed. 1962, New York: W.A. Benjamin, Inc. 97.
16. Kaplan, W., *Ordinary Differential Equations*. 1958.
17. Wegener, W.A. and R. Rigler, *Separation of translational and rotational contributions in solution studies using fluorescence photobleaching recovery*. *Biophysical Journal*, 1984. **46**: p. 787-793.
18. Marquardt, D.W., *An algorithm for least-squares estimation of nonlinear parameters*. *Journal of Social and Industrial Applied Mathematics*, 1963. **11**(2): p. 431-441.

19. Goffe, W.L., G.D. Ferrier, and J. Rogers, *Global optimization of statistical functions with simulated annealing*. Journal of Econometrics, 1994. **60**: p. 65-99.
20. Rahman, N., et al., *Rotational dynamics of type I FcEpsilon receptors on individually-selected rat mast cells studied by polarized fluorescence depletion*. Biophysical Society, 1992. **61**: p. 334-346.
21. Cantor, C.R. and P.R. Schimmel, *Techniques for the study of biological structure and function*, in *Biophysical Chemistry II*, Freeman, Editor. 1980: San Francisco. p. 441.
22. Kask, P., et al., *Fluorescence correlation spectroscopy in the nanosecond time range: rotational diffusion of bovine carbonic anhydrase B*. European Biophysics Journal, 1987. **14**: p. 257-61.
23. Yguerabide, J., H.F. Epstein, and L. Stryer, *Segmental flexibility in an antibody molecule*. J Mol Biol, 1970. **51**(3): p. 573-90.
24. Riddiford, C.L. and B.R. Jennings, *Kerr effect study of the aqueous solutions of three globular proteins*. Biopolymers, 1967. **5**: p. 557-571.
25. Lovejoy, C., D.A. Holowka, and R.E. Cathou, *Nanosecond fluorescence spectroscopy of pyrenebutyrate-anti-pyrene antibody complexes*. Biochemistry, 1977. **16**(16): p. 3668-3672.
26. Chan, L.M. and R.E. Cathou, *The role of the inter-heavy chain disulfide bond in modulating the flexibility of immunoglobulin G antibody*. Journal of Molecular Biology, 1977. **112**(4): p. 653-656.
27. Ferrer, M.L., et al., *The conformation of serum albumin in solution: A combined phosphorescence depolarization-hydrodynamic modeling study*. Biophysical Journal, 2001. **80**(5): p. 2422-2430.

28. Korchuganov, D.S., et al., *Determination of protein rotational correlation time from NMR relaxation data at various solvent viscosities*. Journal of Biomolecular NMR, 2004. **30**(4): p. 431-442.
29. Yao, J., et al., *A Comparative Study of Rotational Relaxations of an Eosin-protein Complex Using Delayed Fluorescence and Phosphorescence*. Journal of Modern Optics, 1992. **39**(11): p. 2363-2373.
30. Priev, A., et al., *Glycerol decreases the volume and compressibility of protein interior*. Biochemistry, 1996. **35**(7): p. 2061-2066.
31. Perrin, F., *Mouvement brownien d'un ellipsoïde (II). rotation libre et dépolarisation des fluorescences. translation et diffusion de molécules ellipsoïdales*. Journal de Physique et le Radium, 1936. **7**(1): p. 1-11.
32. Monkos, K. and B. Turczynski, *Determination of the axial-ratio of globular-proteins in aqueous-solution using viscometric measurements*. International Journal of Biological Macromolecules, 1991. **13**(6): p. 341-344.
33. Squire, P.G., P. Moser, and C.T. O'Konski, *The hydrodynamic properties of bovine serum albumin monomer and dimer*. Biochemistry, 1968. **7**(12): p. 4261-4272.
34. McCoy, L.F. and K.P. Wong, *Renaturation of bovine erythrocyte carbonic anhydrase-B denatured by acid, heat, and detergent*. Biochemistry, 1981. **20**(11): p. 3062-3067.
35. Tilley, L., et al., *Rotational diffusion of human lipoproteins and their receptors as determined by time-resolved phosphorescence anisotropy*. J Biol Chem, 1988. **263**(33): p. 17541-7.

36. Londo, T.R., et al., *Fluorescence depletion measurements in various experimental geometries provide true emission and absorption anisotropies for the study of protein rotation*. Biophysical Chemistry, 1993. **48**(2): p. 241-257.
37. Song, J., et al., *The mast cell function-associated antigen and its interactions with the type I Fcepsilon receptor*. Biochemistry, 2002. **41**(3): p. 881-9.

Chapter V: Overview and Future Directions

In this chapter, I discuss further avenues which could address some of the limitations posed by the methods presented in Chapters II and III involving quantum dots. These include faster cameras which would allow us to access timescales below those which we were able to access with our TTSPC measurements in Chapter III, utilizing gold nanorods instead of quantum dots to improve the signal-to-noise ratio for both of these studies, and a method to obtain a true anisotropy in a microscope system which would represent an improvement over the polarization measurements used in these studies. I also discuss the meaning and broader implications of the studies presented in the preceding chapters.

5.1. Faster Cameras

As previously mentioned in Chapter I, most techniques that have historically been used to measure membrane protein rotation (*e.g.* linear dichroism, E-type fluorescence, time-resolved phosphorescence anisotropy, and fluorescence depletion anisotropy) can only access rotation on timescales of one millisecond or faster. For those techniques reliant on phosphorescence (*e.g.* time-resolved phosphorescence anisotropy and fluorescence depletion anisotropy), the lifetime of organic phosphors imposes a limit on the maximum rotational correlation time that can be recorded.

Approaches combining low-light cameras as we have used in this study with nanoparticle labels (such as the quantum dots used in this study or the gold nanorods discussed in the next

section) can address this limitation associated with the aforementioned techniques as they are capable of measuring rotational correlation times of several milliseconds or slower. In our particular imaging approach, the maximum frame time of our camera imposed a limit on our ability to observe rapid rotation. Even in its fastest mode (540x100 pixels), the Andor EMCCD camera is not capable of measuring rotational correlation times shorter than several times the frame rate of 10 ms. There are currently even faster low-light cameras available, such as the Photometrics “Cascade: 128+”, but this camera can still only acquire a maximum of about 500 frames per second (~1 frame every 2 ms), even if assuming the conservative case of very small images. CMOS cameras with their higher frame rates are still only able to give full-frame images at a frame rate of a few thousand frames per second. Assuming a frame rate of 3,000 frames per second, this would still provide 1 frame every third of a millisecond.. Thus, given that the minimum rotational correlation time accessible to a given camera would have to be at least several times the frame rate, even the fastest low-light cameras cannot be used to measure rotational correlation times of less than about one millisecond and, practically speaking, likely not less than several milliseconds. Thus there is a large window of time too long to be accessed by the aforementioned ensemble measurements (*e.g.* linear dichroism, E-type fluorescence, time-resolved phosphorescence anisotropy, and fluorescence depletion anisotropy), and too short to be examined using imaging approaches employing low-light cameras.

If an imaging method applicable to low-light samples capable of obtaining higher-frame rates could be developed, it could potentially address this issue and enable measurement of rotational correlation times falling within this window spanning about one to several ms. To achieve this, we would propose using a high-speed intensified, gated camera in a “sampling”

mode to examine proteins labeled with nanoparticle probes such as the QDs used in the studies presented here or with the nanorods described in the next section. We believe that the application of such an instrument to imaging measurements in combination with the imaging approaches presented here could represent a potential improvement over an approach relying solely on our imaging method as it could provide supplementary information on these shorter timescales. As far as we are aware, this faster approach to imaging protein rotation is novel and has not been proposed previously by other investigators.

Our lab currently possesses a camera which meets the prior description, the Princeton Instruments 1300YHS-DIF camera system. This is a specialized camera capable of recording images with a variable inter-frame time as low as 200 ns. Using this camera, a split vertically- and horizontally- polarized image of a nanoparticle-labeled protein could first be recorded at some time t and then a second image will be recorded at some later time $t+\tau$ after an inter-frame/lag time of τ . Image pairs could be downloaded from the camera. The product of the spot intensity in the vertically-polarized image and the spot intensity in the horizontally-polarized image of the second image of the pair would give an approximation of the molecule's rotational cross-correlation function at lag time τ . A scan over all possible inter-frame times τ and for all times t would give the true rotational cross-correlation function. With this method, the average time required to download an image from the camera would be significantly less than the magnitudes of typical inter-frame times, and thus it is the inter-frame time that imposes a limit on the lowest correlation time obtainable by this method. If we consider experiments involving quantum dot probes as we have used in this study, for which data acquisition is much simpler than it is for scattering by gold nanorods, it is expected that the rate of photon detection will

impose a limit on the minimum timescales that can be accessed via its effect on the minimum inter-frame time. Considering excitation of QD605 with a 50 mW 532 nm laser, given its molar absorptivity of $5.8 \times 10^5 \text{ L cm}^{-1} \text{ mol}^{-1}$, we expect to record a count rate of 10^5 photons per second per chromophore over an $80 \mu\text{m}$ illuminated field. This could enable measurement of rotational correlation times down to a time less than ten μs . Thus not only would this technique enable measurement of rotational correlation times on timescales between those accessible to TPA and the imaging approach presented in this thesis, but, given its ability to probe timescales overlapping with those capable of being measured by TPA, it could also provide an independent method for measuring fast rotation and perhaps even the hydrodynamic rotation of individual membrane proteins. If we combine this imaging approach using this faster camera with gold nanorod probes, a crucial question will be how to avoid photobleaching and cell photodamage under high laser intensities. To circumvent this issue, we might use only brief laser pulses produced using the modulation input settings on our solid-state lasers. Another possible solution that could accomplish this would be to combine this approach with total internal reflection excitation to reduce penetration into deeper layers of the cell.

5.2. Gold Nanorods

Another potential avenue of future exploration would be to use gold nanorods to conduct similar kinds of studies. Gold nanorods are another type of asymmetric nanocrystal that have been widely used in a variety of analytical, diagnostic, and therapeutic settings [1-3]. Like QDs, gold nanorods also fluoresce and this fluorescence has been monitored in imaging

experiments using 2-photon excitation in a scanning confocal microscope in both solution and cellular preparations [4]. However, this fluorescence does not occur at a rate that could be examined fast enough by imaging. One could, however, make use of the ability of these nanorods to scatter light polarized along their long rod axis. This resonant light scattering could be observed using darkfield microscopy [5]. Like QDs, these nanorods can give large polarized signals and do not bleach under high illumination. In fact, as will be discussed later, they may even be superior with regards to the former.

These nanorods, though larger than QDs, are still unlikely to perturb receptor rotation to a significant extent for the same reasons (their small size and location in the extracellular aqueous medium). Imaging of nanorod samples by electron microscopy suggests a fairly homogeneous size and gives typical nanorod lengths and widths of ~ 35 and ~ 9.5 nm respectively. Their rotational correlation time has been calculated as ~ 3.5 μ s in water.

We have previously synthesized gold nanorods in our lab. These synthesis procedures are based upon work by Sau and Murphy [6] and fine-tuned by Hafner [7], a co-collaborator with our lab. Synthesis of these nanorods involves preparation of nanorod seeds which provide a substrate for subsequent nucleation and growth phases leading to nanorod formation. Experimental conditions are tailored in order to insure production of a homogenous, uniformly-sized nanorod population. This step is typically performed in a 0.1 M cetyltrimethylammonium bromide detergent (CTAB) solution. At this stage, for quality control it is necessary to check both the concentration and size of the nanorods produced. Absorption spectroscopy conducted at the longitudinal plasmon resonance frequency using the large effective molar absorptivity of

$4.4 \times 10^9 \text{ L mol}^{-1} \text{ cm}^{-1}$ can be used to determine nanorod concentration [7]. A couple analytical techniques are available for determining the length of the nanorods. These include imaging by electron microscopy (EM) or via absorption spectroscopy in which a calibration curve generated by plotting the maximum apparent optical absorbance occurring at a λ_{max} of 750 nm of nanorods synthesized with varying reduction times (*i.e.* synthesis times) versus these reduction times. The longer the reduction time, the more time the nanorod has had to grow thus this reduction time is positively correlated with nanorod length. Nanorods of known size and reduction time are then used to construct a calibration curve and the size of the nanorod in question is extrapolated using the calibration curve based on its maximum apparent optical absorbance at its λ_{max} . We have determined the lengths and diameters of nanorods synthesized in our lab with measurements by EM to be 34 ± 3.2 and 9.7 ± 0.8 nm respectively.

After production of a uniformly-sized population of gold nanorods, the nanorod probes are subsequently conjugated to the protein of interest (in this case IgE antibody). Preparation of the nanorods for protein conjugation involves replacement of the CTAB detergent solution with α -thio- ω -carboxypoly(ethylene glycol) and activation with 1-ethyl-[3-dimethylaminopropyl]carbodiimide and N-hydroxysuccinimide. This is followed by quenching with β -mercaptoethanol and conjugation of the protein with the activated nanorod probe.

Though imaging scattering from gold nanorods may be methodologically more complex than imaging fluorescence from QDs, one potential advantage of this alternative approach is a much larger signal to noise ratio. These nanorods have a large effective molar absorptivity $>10^9 \text{ L mol}^{-1} \text{ cm}^{-1}$. Thus the signal produced from resonant light scattering by gold nanorods can be

expected to vastly outweigh background noise from non-resonant light scattering, cellular autofluorescence, *etc.*

5.3. Method to obtain true anisotropy in a microscope

In order to calculate a “true” anisotropy, it is necessary to experimentally obtain information on the intensity of z-polarized light. This information, while accessible to a T-format fluorometer, is typically not accessible to a conventional microscope. To address this issue, we have developed a novel method for obtaining z-polarized emission in a microscope. This method involves recovery of light depolarized by a high numerical aperture objective ($NA \geq 1.3$). For this purpose we have a special mirror located in a reconstructed back focal plane with a suitable hole in the middle of its reflective coating designed to separate light of z-polarized light from that of x- and y- polarized light. Light from the center of the objective is transmitted through the hole and can be directed on a separate optical path to a polarizer which directs this light to two separate detectors. Light from the edge of the objective directed towards the rim of this hole forming an annulus can be directed by a polarizer to a separate detector. This light originating from the edge of the objective and directed towards the rim of the hole contains a substantial component associated with the z-polarized light whereas the former signal originated from the center of the objective and directed towards the center of the hole is primarily comprised of x- and y-polarized light. In this way it is possible to parse out the z-polarized signal from the x- and y- polarized signals. In practice however, this is difficult to implement. Together these approaches could overcome several of the limitations associated with the methods we employed in Chapters II and III of this thesis. Faster cameras could enable us to access faster timescales,

gold nanorods could give an improved signal to noise ratio, and our unique setup could enable us to obtain a true anisotropy in a microscope system.

5.4. Summary of Studies

As discussed in Chapter I, kinetic approaches involving the measurement of diffusion can provide valuable information on molecular size and, by extension, molecular interactions. They can also provide information on the viscosity of the local environment in which a molecule resides. Methods measuring rotational diffusion are superior to those measuring lateral diffusion because rotational diffusion is more sensitive to these variables. In this dissertation, I have presented our studies examining protein rotation both on single cells (Chapters II and III) and in solution (Chapter IV). In Chapters II and III, I presented two approaches to measuring the rotation of the membrane-embedded FcεRI on RBL-2H3 cells in a fluorescence microscope using quantum dots as labels.

In Chapter II, I discussed our fluorescence imaging approach to measuring rotation of the QD-labeled FcεRI. We observed rotational correlation times scattered within a range of ~ 10 - 10^3 ms amongst individual copies of the receptor with weighted geometric mean rotational correlation times within the ~ 100 - 150 ms range. These correlation times are much too long to possibly represent the hydrodynamic rotation of the receptor and, moreover, it is not even theoretically possible to measure the hydrodynamic rotation of the receptor using this technique because any such rotation would happen within one 10 ms frame time. As such, I interpreted these correlation times as representing the rotation of lipid microdomains of varying sizes

containing the QD-labeled receptor. The possibility of the larger complexes formed upon treatment with DNP-BSA being large enough to give correlation on this timescale was considered, but it was thought unlikely that they would be this large. For paraformaldehyde, I expected absolute immobilization of the receptor through intermolecular covalent crosslinking of receptors, but considered the possibility that the lipid microdomains containing them might still rotate on timescales of several hundred milliseconds as was the case for the other treatments. I also examined the effect of cholesterol on rotation via treatment with the cholesterol-depleting agent M β CD. Though it might be expected that the ligand-bound receptor itself would rotate more slowly inside a lipid microdomain which has transitioned to a solid-ordered state, we instead observed a shorter correlation time relative to the other treatments which could have represented the lipid microdomain moving faster upon the lowering of the viscosity of the bulk membrane in which the microdomain was suspended. Because, if anything, I would expect tethering to actin filaments to restrict receptor rotation (though this may not be the case for lateral diffusion), I expected faster rotation upon disruption of these filaments with cytochalasin D. Faster rotation was observed, but this could not correspond to the lone untethered receptor on the long timescales measured by this technique and likely once again corresponds to the rotation of lipid microdomains containing the receptor.

In Chapter III, I discussed our time-tagged single-photon counting approach (TTSPC) to measuring rotation of the QD-labeled Fc ϵ RI. This method enabled us to examine rotation on much shorter timescales fast enough to observe the hydrodynamic rotation of the receptor. We may have accomplished this with the IgE-treated cells as we did observe rotation of a magnitude that could possibly represent this, but unfortunately this result was not reproducible. Though I

expected prolonged correlation times for the DNP-BSA group corresponding to crosslinked receptors, it was difficult to say if there would be any that were small enough to give rotational correlation times below the upper threshold accessible to this technique and the results were once again not reproducible. Paraformaldehyde immobilizes receptors, so any rotation that was observed on this timescale would have had to correspond to the occasional receptor which was not exposed to this agent. M β CD was expected to increase the viscosity of the microdomains in which IgE-bound receptor would be expected to be found, so slower rotation was expected on these short timescales, perhaps in contrast to that observed by our imaging technique. Cytochalasin D was expected to disrupt actin filaments tethered to the receptor which was expected to result in faster rotation. It was difficult to observe such a trend however as the results were not reproducible for this treatment group.

In Chapter IV, we discussed our method for examining protein rotation in solution using continuous fluorescence depletion anisotropy (CFDA). We examined eosin conjugates of carbonic anhydrase, BSA and immunoglobulin G in highly-viscous 90% glycerol solution at various temperatures. We also obtained preliminary cellular results on eosin-IgE-bound Type I Fc ϵ receptors on RBL-2H3 cells and explored how this method might be applied to single cell measurements in a microscope.

5.5. Significance of Studies

Together, these three fluorescence techniques are promising methods to measure rotation of membrane proteins on single cells. The ability to do this in a fluorescence microscope on cells

instead of just with fluorescence spectroscopy on molecules in solution, with single receptor instead of ensemble measurements allowing the distribution of molecular sizes and local environments to be observed, and on multiple different time regimes allowing examination of both small and large structures makes this a versatile combination of methods. As a kinetic approach, rotational measurements, in addition to providing information on the size of molecules and the sizes of higher-order molecular complexes which they form, can provide information on molecular interactions without needing to directly label the molecules which interact with the system under study. This can provide valuable information on the processes in which these molecules participate, especially if they were combined with other approaches to confirm the identity of the binding molecules. As observed in our studies of the FcεRI (the subject of Chapters II and III of this thesis), rotational measurements allowed us to observe receptor crosslinking upon treatment with IgE and DNP-BSA. The crosslinked receptors exhibit much longer rotational correlation times than the un-crosslinked IgE-bound receptor. This simulates the processes that would be expected to occur upon exposure of cells expressing this receptor to foreign antigen. In this manner, our methods can help elucidate the processes occurring on the cell surface after binding to IgE antibody which precede the intracellular events leading to an immune response. It is natural to ask whether these methods could be adapted in order to be applied to other systems. For instance, our lab has extensively studied the luteinizing hormone receptor (LH receptor), a G-protein-coupled receptor (GPCR). In this system, binding of receptor to ligand also initiates receptor-receptor interactions preceding intracellular signaling events. Although the mechanism of receptor aggregation is different for this system and this family of receptors in general, our methods could be extended to such systems and seem promising in their ability to enable us to observe these processes as well.

The rotational information obtained from these approaches can also provide information on the local environment of cell receptors. Rotation is sensitive to changes in local viscosity which can give important information on the surrounding membrane lipid composition and compartmental organization. Our studies have focused on the role of cholesterol on receptor membrane dynamics. These studies improve our understanding of how cholesterol modifies formation and dissolution of lipid microdomains, and how it regulates membrane viscosity both inside these domains and in the larger bulk membrane. Because we have developed approaches capable of examining both short timescales, we are able to observe rotation of individual receptors which can provide information on the immediate environment around the receptor both inside and outside a lipid microdomain. With the ability to access longer timescales, we are also able to examine rotation of the domains themselves, which can provide information about the membrane environment in which the domain containing the receptor is suspended.

5.6. Broader Importance of Work

The cell membrane is the primary theater in which information is transmitted to cells from the outside environment. The standard model for this process involves binding of ligands to integral membrane receptors which triggers a subsequent intracellular signal transduction cascade. However, this simplified conventional model ignores the complex dynamics occurring on the cell surface which precede these intracellular signaling events. A proper understanding of these processes is important as they are a determinant of the extent to which downstream intracellular signaling events occur which ultimately determines cellular responses. As has been

demonstrated in this dissertation, measurements of rotational diffusion provide a viable method for examining these dynamics occurring within the cell membrane.

Because rotational diffusion is sensitive to molecular size, it enables the observation of molecular interactions occurring on the cell surface. For example, in these studies we observed the crosslinking of IgE-bound Fc ϵ RI by DNP-BSA through this method. This process is analogous to what would occur upon crosslinking by foreign antigen. Our methods enabled us to differentiate between receptors exposed to antigen and receptors in the un-crosslinked state in the absence of antigen. It is easy to imagine applying this method to examination of other systems such as G protein-coupled receptors which are a common target for a variety of pharmaceuticals. For example, the luteinizing hormone receptor undergoes oligomerization upon exposure to its cognate ligand luteinizing hormone or the homologous ligand human chorionic gonadotropin. One could imagine the rotational approaches described in this thesis enabling the differentiation between ligand-bound and non-ligand-bound receptors.

Because rotational diffusion is also sensitive to viscosity, measurements of this phenomenon offer the ability to examine changes in the local environment around a protein. This was observed in the CFDA study when we examined the dependence of rotational correlation times of proteins in solution on temperature and viscosity. This was observed with cell receptors by our imaging and TTSPC methods when IgE-bound Fc ϵ RI expressed on RBL-2H3 was treated with M β CD. The slower rotation on short timescales examined by TTSPC was thought to be attributable to the receptor being subjected to a more viscous local environment within a microdomain, consistent with the transition of these microdomains from a liquid-ordered to a

solid-ordered state. The faster rotation on long timescales examined by imaging was thought to be attributable to the microdomain containing the receptor being subjected to a less viscous bulk membrane environment caused by transition of the bulk membrane from a liquid-ordered to a liquid-disordered state. One could imagine using these techniques to observe the localization of receptors to lipid microdomains upon binding of ligand as well, though this would require labeling of the receptor itself instead of the ligand in order to observe rotation of unbound receptors outside of microdomains.

The three methods used to examine rotational diffusion described in this dissertation each have their own unique merits which make them suitable for different specific applications. Both the imaging and TTSPC techniques enable measurement of the rotation of single receptors on single cells in a microscope. The imaging approach allows examination of rotation on long ms timescales which can reveal the origin of the limiting anisotropies observed by ensemble measurements on shorter timescales such as TPA. As we observed here, this technique has the potential to reveal phenomena such as the libration of lipid microdomains containing the receptors or rotation of very large receptor complexes. TTSPC enables examination of shorter μ s timescales and so can reveal hydrodynamic rotation of receptors and small receptor complexes. Thus these two techniques are complementary as they allow access to both short and long timescales and so give a more complete understanding of receptor rotational movement. The CFDA approach takes advantage of the long lifetimes associated with triplet decay along with the large signal given by fluorescence. By exploiting the long lifetimes of phosphors such as eosin and erythrosine, this technique is able to access μ s timescales similar to our TTSPC approach and therefore has the ability to observe hydrodynamic rotation of membrane proteins.

As a proof of principle, it was demonstrated that this spectroscopic technique could be used to examine the dependence of rotational correlation time on the hydrated specific volumes of various proteins in viscous solutions with high concentrations of glycerol. We also showed how it could be adapted to examine rotation of membrane proteins such as FcεRI on single cells and speculated on how this approach could be improved for such systems by adapting it for a microscope system. Combining these different approaches enables a more complete characterization of the rotational behavior of membrane receptors which provides a better understanding of these systems.

5.7. References

1. Ren, X., et al., *Amperometric glucose biosensor based on a gold nanorods/cellulose acetate composite film as immobilization matrix*. *Colloids Surface Biointerfaces*, 2009. **72**: p. 188-192.
2. Chanda, N., et al., *Gastrin releasing protein receptor specific gold nanorods; breast and prostate tumor avid nanovectors for molecular imaging*. *Nano Letters*, 2009. **9**: p. 1798-1805.
3. Dickerson, E., et al., *Gold nanorod assisted near-infrared plasmonic photothermal therapy (PPTT) of squamous cell carcinoma in mice*. *Cancer Letters*. 2008. **269**: p. 57-66.
4. Wang, H., et al., *In vitro and in vivo two-photon luminescence imaging of single gold nanorods*. *Proc Natl Acad Sci USA*, 2005. **102**: p. 15752-15756.
5. Sonnichsen, C. and A. Alivisatos, *Gold Nanorods as novel nonbleaching plasmon-based orientation sensors for polarized single-particle microscopy*. *Nano Letters*, 2005. **5**: p. 301-304.
6. Sau, T. and C. Murphy, *Seeded high yield synthesis of short Au nanorods in aqueous solution*. *Langmuir*, 2004. **20**: p. 6414-6420.
7. Liao, H. and J. Hafner, *Gold nanorod bioconjugates*. *Chemical Materials*, 2005. **17**: p. 4647-4651.

Appendices

Program Code 1: Image Corr Mod15k

```
(* GOAL: assign bkg and g-factor to QD traces *)
(* so as to minimize dependence of r and s *)
(* use the new defn of g as 1+f and 1-f *)
(* mpy 2 reals=~0.4us div 2 reals=~8us *)
(*****
)

(*****
)
(* hard-define column numbers for Excel sheet
*)
(*****
)
ClearSystemCache[];
ClearAll [iFirstRow,iTime,iv,ih]
SetDirectory[NotebookDirectory[]];
inFileName="Ige-1.csv";
inFileName="2015-04-10-DNP-3-LRB.csv";
inFileName="2013-01-14-Ige-11-LRB.csv";
inFileName="Ige-1-LRB.csv";
inFileName="2015-01-06-dry655-6-LRB.csv";
inFileName="2013-01-14-Ige-5-LRB.csv";
inFileName="2015-01-31-Ige-1-LRB.csv";
(*inFileName="pf-3-LRB.csv";*)
(*inFileName="PF-4-LRB.csv";*)
(*inFileName="SimRS-10k-NoBl.csv";*)
(*inFileName="SimRS-100k-NoBl.csv";*)
(*inFileName="SimRS-10k-NoBl-gIs0.csv";*)
(*inFileName="SimRS-100k-SD5-AllZeros.csv";*)
(*inFileName="SimRS.csv";*)
(*inFileName="Ige-1-1.csv";*)
(*inFileName="Ige-1.csv";*)
(*inFileName="SimRS-100k-SD5-AllZeros.csv";*)
(*inFileName="SimRS-100k-e2e1-rTau1-rMax0.csv";*)
(*inFileName="SimRS-100k-s10-e2s100-rTau1-rMax0p0.csv";*)
(*inFileName="SimRS-10k-gmp4-c320-b50-s100-e2s10-exp8-
rTau20-rMax0p5.csv";*)
(*inFileName="SimRS-10k-gmp4-c320-b50-s100-e2s10-exp8-
rTau20-rMax0p5-A.csv";*) (*inFileName="SimRS-10k-gmp4-c320-
b50-s100-e2s10-exp8-rTau20-rMax0p5-E.csv";*)
(*inFileName="SimRS-10k-gmp4-c320-b50-s100-e2s10-tOnOff1-
rTau20-rMax0p5-E.csv";*)
(*inFileName="SimRS-10k-gmp4-c320-b50-s100-e2s10-tOnOff1-
```

```

rTau20-rMax0p5-F.csv";*) (*inFileName="SimRS-10k-gmp4-c320-
b50-s100-e2s10-tOnOff1-rTau20-rMax0p04.csv";*)
(*inFileName="IgE-5-10000-510-LRB.csv";*)
(*inFileName="IgE-1-510-LRB.csv";*)
(*inFileName="dataFile.csv"; (* mean=50, SD=var=1*)*)
(*inFileName="IgE-1-510-QdBkg.csv"; *)
(*inFileName="2015-04-10-IgE-7-4000-LRB.csv";*)

iFirstRow=2;
(*iLastRow=1+1000;*)
iDotInput=9; (* iDot=-1 □ no background traces i.e. old
SimRS *)
iDot=If[StringFreeQ[inFileName,"SimRS",IgnoreCase-
>True]==False,
  1,iDotInput];
If[iDot==-1,
  n2bvc=Input["n2bvc=n2bhc="];
  n2bhc=n2bvc];
iTime=3*iDot+4;
iv=3*iDot+5;
ih=3*iDot+6;
ibv=2;
ibh=3;
base=StringTake[inFileName,StringLength[inFileName]-
4]<>If[iDot!=-1,"-d"<>IntegerString[iDot,10,2],""];

(*****
)
(* set graphics point sizes (fract of total graph)
*)
(*****
)
SetOptions[{Plot,ListPlot},
  PlotStyle->PointSize[0.01],
  PlotStyle->{RGBColor[0.2472,0.24,0.6],
  RGBColor[0.6,0.24,0.4429],
  RGBColor[0.6,0.5470,0.24],
  RGBColor[0.24,0.6,0.3369]},
  BaseStyle->{
  FontFamily->"Helvetica",
  FontWeight->"Bold" },
  ImageSize->{144,108}(*Medium*)
];

(* SZEKELY "DCOR"
*****
(* END SZEKELY "DCOR"

```

```

*****

(*$MinPrecision=31;*)
ClearAll[parListCorrelate]
ClearSystemCache[];
(*Needs["CCompilerDriver`"];
Needs["SymbolicC`"];
$CCompiler =
{"Compiler"□CCompilerDriver`VisualStudioCompiler`VisualStudioCompiler,
"CompilerInstallation"□"c:\\Program Files\\Microsoft Visual Studio 12.0" (*32bit*)};*)

(*$CCompiler =
{"Compiler"□CCompilerDriver`VisualStudioCompiler`VisualStudioCompiler,
"CompilerInstallation"□"c:\\Program Files (x86)\\Microsoft Visual Studio 12.0" (*64bit*)};*)

(* (*****
*****
*)
(* important relations
*)
(*****
*****
*)
e2r=(4 v^2 dh^2)/(h+v)^4+(4 h^2dv^2)/(h+v)^4-(8 (h v) dh
dv)/(h+v)^4; (*OK*)
e2r=1/s^2((1+r)^2 dh^2+(1-r)^2 dv^2 -2(1-r^2)dv dh);
e2d=dh^2+dv^2- 2 dhdv;
e2s=dh^2+dv^2+ 2 dhdv;
eds=dv^2- dh^2;
(* note that for corr'd bkgd, v~h so dhdv~0*) *)

(*****
*****
*)
(* calc variance of ratio
*)
(*****
*****
*)
(*ClearAll[drsqr,dd,ds,s,d,v,h,dv,dh];
drsqr=((v-h+dv-dh)/(v+dv+h+dh)-(v-h)/(v+h))^2
Series[drsqr, {dv,0,4} ,{dh,0,4}]
(-((-h+v)/(h+v))+(-dh+dv-h+v)/(dh+dv+h+v))^2
((4 v^2 dh^2)/(h+v)^4-(8 v^2 dh^3)/(h+v)^5+(12 v^2
dh^4)/(h+v)^6+O[dh]^5)+(-((8 (h v) dh)/(h+v)^4)+(16 h
v)/(h+v)^5-(8 v^2)/(h+v)^5) dh^2+(-((24 h v)/(h+v)^6)+(24

```

```

v^2)/(h+v)^6) dh^3+((32 h v)/(h+v)^7-(48 v^2)/(h+v)^7)
dh^4+O[dh]^5) dv+((4 h^2)/(h+v)^4-(8 (h^2-2 h v)
dh)/(h+v)^5+(12 (h^2-4 h v+v^2) dh^2)/(h+v)^6-(16 (h^2-6 h
v+3 v^2) dh^3)/(h+v)^7+(20 (h^2-8 h v+6 v^2)
dh^4)/(h+v)^8+O[dh]^5) dv^2+(-(8 h^2)/(h+v)^5)+(24 (h^2-h
v) dh)/(h+v)^6-(16 (3 h^2-6 h v+v^2) dh^2)/(h+v)^7+(80
(h^2-3 h v+v^2) dh^3)/(h+v)^8-(120 (h^2-4 h v+2 v^2)
dh^4)/(h+v)^9+O[dh]^5) dv^3+((12 h^2)/(h+v)^6-(16 (3 h^2-2
h v) dh)/(h+v)^7+(20 (6 h^2-8 h v+v^2) dh^2)/(h+v)^8-(120
(2 h^2-4 h v+v^2) dh^3)/(h+v)^9+(140 (3 h^2-8 h v+3 v^2)
dh^4)/(h+v)^10+O[dh]^5) dv^4+O[dv]^5
PDF[BinormalDistribution[{Subscript[μ, 1],Subscript[μ,
2]},{Subscript[σ, 1],Subscript[σ, 2]},{ρ},{x,y]}]

```

$$\frac{1}{\sigma_1 \sigma_2 \sqrt{1 - \rho^2}} \exp\left\{-\frac{1}{2(1 - \rho^2)} \left[\frac{(x - \mu_1)^2}{\sigma_1^2} - 2\rho \frac{(x - \mu_1)(y - \mu_2)}{\sigma_1 \sigma_2} + \frac{(y - \mu_2)^2}{\sigma_2^2} \right]\right\}$$

```


```

```

SubscriptBox[σ, 1]\
SqrtBox[
FractionBox[1,
SubsuperscriptBox[σ, 2, 2]]\
SubsuperscriptBox[σ, 2, 3]σ, σ
SqrtBox[1 -
SuperscriptBox[ρ, 2]]\
SqrtBox[-
FractionBox[1, σ(-1 +
SuperscriptBox[ρ, 2])]\
SubsuperscriptBox[σ, 1, 2]]], Assumptions {
Re[
FractionBox[1, σ(-1 +
SuperscriptBox[ρ, 2])]\
SubsuperscriptBox[σ, 1, 2]] < 0 && Re[
FractionBox[1,
SubsuperscriptBox[σ, 2, 2]] > 0 &&
SubsuperscriptBox[σ, 2, 2] > 0 &&
SuperscriptBox[ρ, 2] > 0 &
SubsuperscriptBox[σ, 1, 2] > 0}]\n
SqrtBox[
FractionBox[1, 1 -
SuperscriptBox[ρ, 2]]\
SqrtBox[1 -
SuperscriptBox[ρ, 2]]\ σ(1 + 2\
SuperscriptBox[ρ, 2])\ Abs[
SubscriptBox[σ, 1]]\
SubscriptBox[σ, 1]]\
SqrtBox[
FractionBox[1,
SubsuperscriptBox[σ, 2, 2]]\
SubsuperscriptBox[σ, 2, 3]\n\ σ(1 + 2\
SuperscriptBox[ρ, 2])\
SuperscriptBox[
SubscriptBox[σ, 1], 2]
SubsuperscriptBox[σ, 2, 2]]
*)

(* (*****
*****
*)
(* variance of product and quotient
*)
(* (*****
*****
*)
ClearAll[d,e,s,t,w];
w=Expand[((d+e)/(s+t)-d/s)^2];
Series[s^4 w,{e,0,2},{t,0,2}];

```

```

(*red=terms lost on integ*)
(d^2 t^2)/s^4-1/s^42 (d s) t e+(4 d t^2e)/s^4+(s^2e^2)/s^4-
(2 s t e^2)/s^4+(3 t^2 e^2)/s^4;
(* for normal distn's,  $\Sigma t^2 e^2 = 2eds^2+e2s*e2d$  *)
ClearAll[d,e,s,t,w];
w=Expand[((d+e)(s+t)-d*s)^2];
(*red=terms lost on integ*)
e^2 s^2+2 d e s t +2 e^2 s t +d^2 t^2 +2 d e t^2+e^2 t^2;*
(* for normal distn's,  $\Sigma t^2 e^2 = 2eds^2+e2s*e2d$  *)

(* *****
***** *)
(* series weighting of anis correlation products
*)
(* *****
***** *)ClearAll[r1,r2,v1,v2,iOrder,ir1,ir2];
nMax=1
q=1/(v1*v2)Series[1/(1+r1/v1+r2/v2),{r1,0,nMax},{r2,0,nMax}
];
Do[Do[
ir2=iOrder-ir1;
c=SeriesCoefficient[q,{ir1,ir2}]*r1^(2ir1)*r2^(2 ir2);
Print["r1=",ir1," r2=", ir2, " c=",c]
,{ir1,0,iOrder}},{iOrder,0,nMax}]*)

(* *****
***** *)
(* test autocorrelation for finite data set using
LISTCORRELATE *)
(* *****
***** *)
ClearAll[DataCorrelate];
DataCorrelate[t_,u_]:=Module[{n},
z=ListCorrelate[t,u,1,0];n=Length[t];Table[z[[i]]/(n-
i+1),{i,n}]];
(*t={4,3,2,1}
u={1,2,3,4}
z=ListCorrelate[t,u,1,0]
z1=Table[z[[i]]/(Length[z]-i+1),{i,Length[z]}]
DataCorrelate[t,u]*)

(* *****
***** *)
(* define pattern replace to pull constants out of sums
*)
(* *****
***** *)

```

```

(* patt /; test is a pattern which matches only if the
evaluation of test yields True *)
(* lhs  $\square$  rhs "\\\"\\\"\\\"RuleDelayed\\\"\\\"\\\"\\\"\" represents a
rule that transforms lhs to rhs evaluating rhs only after
the rule is used *)
(* FreeQ[expr,form] yields True if no subexpression in expr
matchesformand yields False *)
ClearAll[BringOut,outrules];
outrules={Sum[f_+
g_,it:{x_Symbol,__}]:>Sum[f,it]+Sum[g,it],Sum[c_
f_,it:{x_Symbol,__}]:>c
Sum[f,it]/;FreeQ[c,x],Sum[c_,it:{x_Symbol,__}]:>c
Sum[1,it]/;FreeQ[c,x]];
BringOut[s_]:=s //. outrules
(*BringOut [Sum [c*i*x[[i]]1,{i,n}]];*

(*****
*****
(* map Sum function over sum of terms
*)
(*****
*****
(*ClearAll[x,y,z,qd0,n];
qd0= x[[i]] +2y[[i]]^2+ 3 z[[i]]^3;
gd0=Total[Sum[#, {i,1,n}]&/@(List@@qd0)];
BringOut[gd0]*)

(*****
*****
(* weighted covariance function
*)
(*****
*****
ClearAll[wtdCov];
wtdCov[f_,g_,w_]:=Module[{n,wTot,temp},
n=Length[f];
wTot=Total[w];
temp=Total[f*g*w]/wTot-Total[f*w]/wTot Total[g*w]/wTot];
(* fails for f=g and i=1 *)

(*****
*****
(* weighted correlation coeff
*)
(*****
*****
ClearAll[wtdCorrCoeff];

```

```

wtdCorrCoeff[f_,g_,w_]:=Module[
  {},
  wtdCov[f,g,w]/ $\sqrt{\text{wtdCov}[f,f,w]*\text{wtdCov}[g,g,w]}$ 
]

(*****
*****
(* abbreviate FortranForm for printing
*)
(*****
*****
ClearAll[ff];
ff[x_]:=FortranForm[x];

(*****
*****
(* define gRound to get suitable form printing large,
small nums *)
(*****
*****
ClearAll[gR];
gR[x_,sf_]:= (* "g Round" *)Module[{e,xr},
  (e=MantissaExponent[x][[2]]-sf;
  xr=(10^e)*Round[x*10^(-e)];
  xr)];

(*****
*****
(* define gF "George Format" for compact 1-line output
*)
(*****
*****
ClearAll[gF];
gF[x_]:=ff[gR[x,6]];

(*****
*****
(* define "appendCol"
*)
(* does NOT expand table if col too long or pad col if too
short *)
(*****
*****
appendCol[a_,b_]:=(*Module[{aa},If[Length[Dimensions[x]]>1,
aa={a};Transpose[Append[aa,b]],
Transpose[Append[Transpose[a],b]]];*)

```

```

If[Length[a]!=Length[b],Print["Unequal lengths. Unable to
append"];a,
If[Length[Dimensions[a]]==1,
Join[Transpose[{a}], Transpose[{b}],2],
Join[a, Transpose[{b}],2]]

(*****
*****
(*  define "addParam"
*)
(*  if new var runs below existing cols, table is padded
line of blanks*)
(*****
*****)
ClearAll[addParam];
addParam[a_,nameCol_,valCol_,name_,val_]:=
Module[{nParams,ap},
nParams=0;
Do[If[a[[i,nameCol]]!=
"",nParams=nParams+1],{i,2,Length[a]}];

If[Length[a]==nParams+1,ap=Append[a,Table["",{i,Length[a][[1
]]}]]];,ap=a];
ap=ReplacePart[ap,{nParams+2,nameCol}-> name];
ap=ReplacePart[ap,{nParams+2,valCol}-> val];
ap
]

(*****
*****
(*  define "mlpNeg" ==-Log prob of getting negative v or h
*)
(*****
*****)
ClearAll[mlpNeg];
mlpNeg[b_,sb_,v_]:=Log[2]-Log[Erfc[(b-v)/(√2 sb)]];
(*****
*****)
(*  extrapolate xc or ac curve to time zero
*)
(*****
*****)
extrap[x_]:=x[[1]],x[[1]]-x[[2]],x[[1]]-2x[[2]]+x[[3]]];

(*****
*)
(*  get v,h from CSV file

```

```

*)
(* iDot=-1 □ no background traces
*)
(*****
*)
ClearAll[n,v,h]
data=Import[inFileName,"CSV"];
lastData=Length[data];
time=data[[iFirstRow;;(*iLastRow*),iTime]]; (*1st dim is
DOWN, 2nd ACROSS *)
v=data[[iFirstRow;;(*iLastRow*),iv]]; (*starts 2nd row,
indicated col *)
h=data[[iFirstRow;;(*iLastRow*),ih]]; (*starts 2nd row,
indicated col *)
If [iDot!=-1,
  bv=data[[iFirstRow;;(*iLastRow*),ibv]]; (*starts 2nd row,
indicated col *)
  bh=data[[iFirstRow;;(*iLastRow*),ibh]],
  (*else*)
  bv=Table[0,{lastData} ];
  bh=Table[0,{lastData}]]; (*starts 2nd row, indicated col
*)
Print["inFileName=",n=Length[time]];

(*****
****)
(* examine raw data
*)
(*****
****)
  bvf=bv-Mean[bv];  bhf=bh-Mean[bh];
gbv=DataCorrelate[bvf,bvf];
gbh=DataCorrelate[bhf,bhf];
inputPlots={
  vhRawPlot=ListPlot[{v,h},PlotLabel->"vhRaw",PlotRange-
>{0,1.25Max[v,h]}],
  bkgRawPlot=ListPlot[{bv,bh},PlotLabel-
>"bkgsRaw",PlotRange-
>{0,1.25Max[v,h]}],bkgCorrPlot=ListPlot[{gbv,gbh},PlotLabel-
->"bkgCorr"]}

(*****
****)
(* set date-time format to distinguish file versions
*)
(*****
****) $DateStringFormat={"Year", (*"-",*) "Month", (*"-

```

```

", *) "Day", "-", "Hour24", "Minute", "Second"};

(*****
*****)
(* generate timestamp for all output this pass
*)
(*****
*****)
tdList=DateList[];
dateFormat={"Year", "-", "Month", "-", "Day"};
timeFormat={"Hour", ":", "Minute", ":", "Second"};
timeStampFormat={"Year", (*"-", *) "Month", (*"-", *) "Day", "-
", "Hour24", "Minute", "Second"};
dateString=DateString[tdList, dateFormat];
timeString=DateString[tdList, timeFormat];
timeStampString=DateString[tdList, timeStampFormat];

(*****
*****)
(* generate parallelizable ListCorrelate
*)
(*****
*****) (*parListCorrelate=Compile[{{ker, _Real, 1}, {list,
_Real, 1}, {klist, _Integer}, {p,
_Real}}, ListCorrelate[ker, list, klist, p]
, CompilationTarget->"C", RuntimeAttributes->{Listable}, Paralle
lization->True];*)
(*parListCorrelate[ker_, list_, klist_, p_] := ListCorrelate[ker
, list, klist, p];*)

(*****
*****)
(*ANIS: fit single exponential decay
*)
(*****
*****)
ClearAll[fitExpDecay];
fitExpDecay[fitData_,
fitDataName_, wtData_, nEff_, kFirst_, kLast_, fitPlotName_] :=
Module[
{gTable, wTable, iPass, g0, g $\infty$ , gTau, t},
If[kLast > nEff, Print["kLast=", kLast, " > nEff=", nEff, ".
Results unpredictable"]];
gTable=Table[{i-1, fitData[[i]]}, {i, nEff}];
wTable=Table[wtData[[i]], {i, nEff}];
(*gTable=Table[{i-
1, RandomVariate[NormalDistribution[0, 2]]}, {i, nEff}];

```

```

wTable=Table[1,{i,nEff}];*)
Do[If[k<kFirst||k>kLast,wTable[[k]]=0},{k,nEff}];
iPass=0;

Monitor[
  nlm=NonlinearModelFit[
    gTable,
    g $\infty$ +(g0-g $\infty$ )*Exp[-t/gTau],
    {{g0(*,.0005*)},{g $\infty$ (*,-.0005*)},{gTau}},
    t,
    Weights->wTable,
    StepMonitor :>{iPass=iPass+1},
    VarianceEstimatorFunction->(Total[#12 #2]/Total[#2]&),
    MaxIterations->1000(*Infinity*)
  ],
  Pause[0.0];
  "iPass="<>ToString[gF[iPass]]<>"
chiSq="(*<>ToString[gF[redChiSq]]*)<>"
tDecay="<>ToString[gF[gTau]]<>"
gInf="<>ToString[gF[g0]]<>ToString[gF[g $\infty$ ]]<>" gAmp="
  ];

Print["\n","FIT RESULTS"];
paramTable=nlm["ParameterTable"];
Print["params=",params=nlm["BestFitParameters"]];
Print["errs=",errs=nlm["ParameterErrors"]];
Print["estdSD=",estdSD= $\sqrt{\text{nlm["EstimatedVariance"]}}$ ];
Print["nlm[0]=",nlm[0]];

params=nlm["BestFitParameters"];errs=nlm["ParameterErrors"];
estdVar=nlm["EstimatedVariance"];gSD= $\sqrt{\text{estdVar}}$ ;

gZero=params[[1,2]];gInf=params[[2,2]];gTd=params[[3,2]];
gInfErr=errs[[1]];gZeroErr=errs[[2]];gTdErr=errs[[3]];
fluctCalc=Table[nlm[i-1],{i,1,nEff}];

xyCalc=Table[{i-1,nlm[i-1]},{i,kLast}];range={1.1Min[xyCalc[[All,2]],gTable[[All,2]],1.1*Max[xyCalc[[All,2]],gTable[[All,2]]]};

fitPlot=ListPlot[{gTable[[kFirst;;kLast,2]],xyCalc[[kFirst;;kLast,2]]},PlotLabel->fitPlotName];
];

(*****

```

```

****)
(*  define "marq"
*)
(*****
****)
(*nSum=1;
marq[
v_List,h_List,iWt_Integer,iFirst_Integer,iLast_Integer,kFir
st_Integer,kLast_Integer,a1_Real,a2_Real,a3_Real]
:=Sum[depend[v,h,iWt,iFirst,iLast,kFirst,kLast,a1,a2,a3]^2,
{i,,nSum}];*)

(*
(*****
****)
(*  define "depend"
*)
(*****
****)

ClearAll[depend];
depend=Compile[{{v,_Real,1},{h,_Real,1},{iWt,_Integer},
{iFirst,
_Integer},{iLast,_Integer},{kFirst,_Integer},{kLast,_Intege
r},{g,_Real},{c,_Real},{b,_Real},{aIn,_Real}},*)

ClearAll[depend];

depend[v_List,h_List,iWt_Integer,iFirst_Integer,iLast_Integ
er,kFirst_Integer,kLast_Integer,nBkgsDummy_Integer,gDummy_R
eal,cDummy_Real,bDummy_Real,fDummy_Real,aIn_Real,iThreshMod
e_Integer,thresh_Real,iInitDummy_Integer,iOrderDummy_Intege
r]:= (

(*****
****)
(*  check input data
*)

(*****
****)
If [Length[v]!= Length[h], Print["v and h unequal
length. Exiting..."];Abort[] ];
nMax=Length[v];
n=If [iLast==0,Length[v],iLast];
m=If [iFirst==0,1,iFirst];

```

```

nEff=n-m+1;
nCorr=nEff;

kMax=If[kLast==0, nEff-1, kLast];
If [kLast>nEff,Print["kMax exceeds nEff.
Exiting..."];Abort[]];
kMin=If[kFirst==0, 2,kFirst];
nk=kMax-kMin+1;

nBkgs=If [nBkgsDummy==0,4,nBkgsDummy];
alpha=If [aIn==0,2.0,aIn]; (* 0<aIn<2; 1=Szekaly;
2=Pearson cov *)
(*f=If [fDummy<=0,0.0,fDummy];*)
(*Print["m=",m," n=",n];*)

(*****
*****
(* set constants
*)

(*****
*****
g=gDummy;
c=cDummy;
b=bDummy;
f=fDummy;
iInit=iInitDummy;
iOrder=iOrderDummy;

(*****
*****
(* create tables
*)
(* use length of v or h as nMax
*)

(*****
*****

createArrays=AbsoluteTiming[
ClearAll[];
scr1D=Table[0,{nEff}];
scr1Dp1=Table[0,{nEff+1}];

(* saved variables and products *)

```

```

vs=scr1D; (* s variance *)
vd=scr1D; (* d variance *)
vr=scr1D; (* r variance *)

ws=scr1D; (* s wt *)
wd=scr1D; (* d wt *)
If[iInit== 1,wr=scr1D;wrSq=Scr1D;wrCb=Scr1D]; (* r wt
*)

(* terms for s *)
sCum=scr1Dp1;
sAvg1=scr1D;
sAvg2=scr1D;

sSq=scr1D;
sSqCum=scr1Dp1;
sSqAvg1=scr1D;
sSqAvg2=scr1D;

sCb=scr1D;
sCbCum=scr1Dp1;
sCbAvg1=scr1D;
sCbAvg2=scr1D;

swr=scr1D;
swrCum=scr1Dp1;
swrAvg1=scr1D;
swrAvg2=scr1D;

sSqwr=scr1D;
sSqwrCum=scr1Dp1;
sSqwrAvg1=scr1D;
sSqwrAvg2=scr1D;

sCbwr=scr1D;
sCbwrCum=scr1Dp1;
sCbwrAvg1=scr1D;
sCbwrAvg2=scr1D;

(* terms for d *)
dCum=scr1Dp1;
dAvg1=scr1D;
dAvg2=scr1D;

dSq=scr1D;
dSqCum=scr1Dp1;
dSqAvg1=scr1D;

```

```

dSqAvg2=scl1D;

dwr=scl1D;
dwrCum=scl1Dp1;
dwrAvg1=scl1D;
dwrAvg2=scl1D;

dSqwr=scl1D;
dSqwrCum=scl1Dp1;
dSqwrAvg1=scl1D;
dSqwrAvg2=scl1D;

(* terms for r *)
wrCum=scl1Dp1;
wrSum1=scl1D;
wrSum2=scl1D;

If[iInit== 1,rwr=scl1D;rSqwr=scl1D; rCbwr=scl1D];
rwrCum=scl1Dp1;
rwrAvg1=scl1D;
rwrAvg2=scl1D;

r=scl1D;
rSq=scl1D;
rSqwr=scl1D;
rSqwrCum=scl1Dp1;
rSqwrAvg1=scl1D;
rSqwrAvg2=scl1D;

rCb=scl1D;
rCbwr=scl1D;
rCbwrCum=scl1Dp1;
rCbwrAvg1=scl1D;
rCbwrAvg2=scl1D;

(* add'l terms for s r *)
r1s1=scl1D;
r2s1=scl1D;
r1s2=scl1D;
r3s1=scl1D;
r2s2=scl1D;
r1s3=scl1D;

r1s1wr=scl1D;
r2s1wr=scl1D;
r1s2wr=scl1D;
r3s1wr=scl1D;

```

```

r2s2wr=scr1D;
r1s3wr=scr1D;

(*r2s1wrSq=scr1D;
r3s1wrCb=scr1D;
r2s2wrSq=scr1D;*)
th=scr1D;

corrResults=Table[scr1D,{32}]
]; (*end create arrays*)

(*****
*****
  (* CALCULATION STARTS HERE
*)

(*****
*****

(*****
*****
  (* input raw data
*)

(*****
*****
  veff=v[[m;;n]];
  heff=h[[m;;n]];
  bveff=bv[[m;;n]];
  bheff=bh[[m;;n]];

(*****
****)
  (* get raw data avgs, var's ....
*)

(*****
****)
  vAvg=Mean[veff];
  hAvg=Mean[heff];
  vVar=Variance[veff];
  hVar=Variance[heff];
  vCov=Covariance[veff,heff];
  e2v=vVar; (* var this v-trace *)

```

```

e2h=hVar; (* var this h-trace *)
e2vh=vhCov; (* var this h-trace *)

(* (*****
*****)
    (* and for g-corrected vc, hc
*)

(*****
*****)
    gFact=(1+g^2);
    vcAvg=vAvg*(1-g);
    hcAvg=hAvg*(1+g); (* cov this v,h-traces *)
    e2vc=e2v*gFact; (* var this v-trace *)
    e2hc=e2h*gFact; (* var this h-trace *)
    e2vhc=e2vh*gFact; (* cov this v,h-traces *)
    *)

(*****
*****)
    (* get baseline variances
*)

(*****
*****)
    (* nBkgs = number of baselines averaged *)
    bvAvg=Mean[bveff];
    bhAvg=Mean[bheff];
    bvVar=Variance[bveff];
    bhVar=Variance[bheff];
    bvhCov=Covariance[bveff,bheff];
    e2bv (*var 1 bkg trace*)=bvVar>(*4 traces*)/nBkgs(*4*);
    e2bh=bhVar/nBkgs; (* var one h-bkg trace *)
    e2bvh=bvhCov/nBkgs; (* cov v,h-bkg traces *)

(*****
*****)
    (* assume any slow BKG fluctuations 100% correlated v $\square$  h
*)
    (* scaled ABOVE for 4 (or other) bkg traces
*)

(*****
*****)

```

```

If[ iDot!==-1, (*if NOT a simulation*)
  rf= $\sqrt{e2bv / e2bh}$  ; (* ratio to simplify calcns below *)
  gEstBkgs=(rf-1)/(rf+1); (* g est'd from bkgs *)
  n2bv=e2bv-rf*e2bv; (* true var in raw v - inten
flucts taken out *)
  n2bh=e2bh-e2bv/rf;
  n2bvc=(1-gEstBkgs)2 n2bv; (* true var in vc *)
  n2bhc=(1+gEstBkgs)2 n2bh;
  n2bsc=n2bvc+n2bhc;
  n2bdc=n2bvc+n2bhc];

(*****
****)
  (* assume any slow DATA fluctuations 100% correlated v
h *)

(*****
****)
  If[ iDot!==-1, (*if NOT a simulation*)
    rf= $\sqrt{e2v / e2h}$  ; (* ratio to simplify calcns below *)
    gEst=(rf-1)/(rf+1); (* g est'd from bkgs *)
    n2v=e2v-rf*e2v; (* true var in raw v - inten fluct
taken out *)
    n2h=e2h-e2v/rf;
    n2vc=(1-gEst)2 n2v; (* true var in vc *)
    n2hc=(1+gEst)2 n2h;
    n2sc=n2vc+n2hc;
    n2dc=n2vc+n2hc];

  (*gbv=DataCorrelate[bv-bvAvg,bv-bvAvg];
gbh=DataCorrelate[bh-bhAvg,bh-bhAvg];
gbvbh=DataCorrelate[bv-bvAvg,bh-bhAvg];*)

(*****
****)
  (* calculate functions
*)
  (* generate subarrays from full input data as needed
*)
  (* COULD replace const 'cBkg' with vaying 'c' *)

(*****
****)

```

```

fillArrays=AbsoluteTiming[
  If [iDot!=-1,
    cBkg=319.75; (*cBkg=c;*)
    (*cBkg=f;*)
    iv=(bveff-cBkg)/(bvAvg-cBkg);
    ih=(bheff-cBkg)/(bhAvg-cBkg);
    it=((bveff-cBkg)+(bheff-cBkg))/((bvAvg-cBkg)+(bhAvg-
cBkg))(*it should=iv,ih, so calc unbiased it*),
    (*else*)
    iv=Table[1,{nEff}];
    ih=Table[1,{nEff}];
    vc=(1-g)(veff-bveff)/it-b;
    hc=(1+g)(heff-bheff)/it-c;
    s=vc+hc;
    d=vc-hc;
    vMin=Min[veff];
    hMin=Min[heff];
    (*Print["s,d calc'd"];*)

  (*****
  *****)
  (* calculate autocorrelation of raw data
  *)

  (*****
  *****)
  (*gvvInit=DataCorrelate[v,v];
  ghhInit=DataCorrelate[h,h];
  gssInit=DataCorrelate[s,s];*)
  gddInit=DataCorrelate[d,d];

  (*****
  *****)
  (* replace variance of raw data with g(0)-g(1)
  *)
  (* this is to estimate counting errors w/o true var in
  st) *)

  (*****
  *****)

  (*e2vc=(1-g)^2extrap[gvvInit][[2]];
  e2hc=(1+g)^2extrap[ghhInit][[2]];
  e2vc=(1-g)^2Variance[bv];
  e2hc=(1+g)^2Variance[bh];*)

```

```

    (*e2s=Variance[s];
    e2d=Variance[d];*)
    (*e2d=extrap[gddInit][[2]];*)
    (*ws0=1/e2s;
    wd0=1/e2d;*)

(*****
*****
    (* calc variances of v,h,s,d INCLUDING blinking
*)

(*****
*****
    e2vc=Variance[vc];
    e2hc=Variance[hc];
    e2vhc=Covariance[vc,hc];
    e2sc=Variance[s];
    e2dc=Variance[d];

(*****
*****
    (* loop through data and generate s,d,r and calc wts
*)
    (* generates sSq,r,vr,wr,qr
*)

(*****
*****
    threshLevel=If [ iThreshMode==0,
        (*0=n2sc*) thresh* $\sqrt{2 * n2sc}$  ,
        (*<>0=abs*) thresh];
    Do[ (*pts i*)
        s0=s[[i]];
        If [s0<threshLevel, nCorr=nCorr-1];
        d0=d[[i]];
        dSq[[i]]=d02;
        r0=If[s0==0.0,0.0,d0/s0];
        r[[i]]=r0;

        s20=s02;
        s30=s20*s0;
        sSq[[i]]=s20;
        sCb[[i]]=s30;

```

```

r20=r02;
r30=r20*r0;
rSq[[i]]=r20;
rCb[[i]]=r20*r0;

vr0=(1+r02)/s20; (* must be mult'd by "e2vc" or "e2hc"
*)
vr[[i]]=vr0;
wr0= $\sqrt{1/vr0}$  ; (* var r0 inverse *)
If[iInit==1, (* if first pass, generate wts. If not,
carry over*)
wr[[i]]=wr0;
rwr[[i]]=r0*wr0;
swr[[i]]=s0*wr0;
dwr[[i]]=d0*wr0;
rSqwr[[i]]=r20*wr0;
rCbwr[[i]]=r30*wr0;

r1s1[[i]]=d0;
r2s1[[i]]=r20*s0;
r1s2[[1]]=r0*s20;
r3s1[[i]]=r30*s0;
r2s2[[i]]=r20*s20;
r1s3[[i]]=r0*s30;

r1s1wr[[i]]=d0*wr0; (*r1s1=d0;*)
r2s1wr[[i]]=r20*s0*wr0;
r1s2wr[[1]]=r0*s20*wr0;
r3s1wr[[i]]=r30*s0*wr0;
r2s2wr[[i]]=r20*s20*wr0;
r1s3wr[[i]]=r0*s30*wr0;
];
rMax=.04;
(*rMax=1;*)
th[[i]]=Re[(1/2)ArcSin[r0/rMax]]

,{i,1,nEff} (* pts i *)
];

(*****
*****
(* generate sNew = 0.5*(s[[i-1]]+s[[i+1]])
*)

(*****
*****

```

```

(*Do[ (*i=2,nEff-1*)
sNew[[i]]=0.5*(s[[i-1]]+s[[i+1]]),
{i,2,nEff-1}
];*)

(*****
*****)
(* normalize weights if desired
*)

(*****
*****)
wrFact=nEff/Total[wr];
wrn=wr*wrFact;

(*****
*****)
(* calc r,s averages
*)

(*****
*****)
sAvg=Total[s]/nEff;
dAvg= Total[d]/nEff;
dSqAvg=Total[dSq]/nEff;
dAvgWtd=Total[r1s1wr]/Total[wr];
rwrAvg=Total[rwr]/Total[wr];

(*****
*****)
(* loop through data and generate everything else
*)

(*****
*****)

Do[ (*pts i*)
(* generate saved terms *)
(* SAVE QR ARRAYS----DON'T SAVE WR ARRAYS *)

ip1=i+1;
wr0=wr[[i]];
wrCum[[ip1]]=wrCum[[i]]+wr0;

```

```

If [iOrder]>=2,
  (* terms for ss *)
  sCum[[ip1]]=sCum[[i]]+s[[i]];
  swrCum[[ip1]]=swrCum[[i]]+swr[[i]];
  sSqCum[[ip1]]=sSqCum[[i]]+sSq[[i]];
  sCbCum[[ip1]]=sCbCum[[i]]+sCb[[i]];

  dCum[[ip1]]=dCum[[i]]+d[[i]];
  dwrCum[[ip1]]=dwrCum[[i]]+dwr[[i]];
  dSqCum[[ip1]]=dSqCum[[i]]+dSq[[i]];

  rwrCum[[ip1]]=rwrCum[[i]]+rwr[[i]];
  rSqwrCum[[ip1]]=rSqwrCum[[i]]+rSqwr[[i]];
  rCbwrCum[[ip1]]=rCbwrCum[[i]]+rCbwr[[i]] (*;

  r1s1wrCum[[i]]=d[[i]]*wr0; (*r1s1=d0;*)
  r2s1wrCum[[i]]=rSq[[i]]*s[[i]]*wr0;
  r1s2wrCum[[1]]=r[[i]]*sSq[[i]]*wr0;
  r3s1wrCum[[i]]=rCb[[i]]*s[[i]]*wr0;
  r2s2wrCum[[i]]=rSq[[i]]*sSq[[i]]*wr0;
  r1s3wrCum[[i]]=r0*s30*wr0*)
]
,{i,1,nEff} (* pts i *)
]; (* end fillArrays*)

```

```

(*****
***)
  (* here calc e2r
*)

```

```

(*****
***)
  (*grrInit=DataCorrelate[rwr,rwr];
  rSqAvg=grrInit[[1]]/(wrCum[[nEff+1]]-wrCum[[1]]);
  rAvg=(rwrCum[[nEff+1]]-rwrCum[[1]])/(wrCum[[nEff+1]]-
wrCum[[1]]);
*)

```

```

(*****
***)
  (* get most significant averages
*)

```

```

(*****
***)

(*****
***)
  (* get region averages
*)

(*****
***)
  regionAvgs=AbsoluteTiming[

    Do[(*correlation time k-1 *)
      iLimit=nEff-k+2; (*add 1 pt since cums start at 1*)
      jLimit=nEff+1;
      nRegionPts=nEff-k+1;

      If[iOrder>=2,
        wrSum1[[k]]=wrCum[[iLimit]]-wrCum[[1]];
        wrSum2[[k]]=wrCum[[jLimit]]-wrCum[[k]];

        sAvg1[[k]]=(sCum[[iLimit]]-sCum[[1]])/nRegionPts;
        sAvg2[[k]]=(sCum[[jLimit]]-sCum[[k]])/nRegionPts;

        swrAvg1[[k]]=(swrCum[[iLimit]]-
swrCum[[1]])/wrSum1[[k]];
        swrAvg2[[k]]=(swrCum[[jLimit]]-
swrCum[[k]])/wrSum2[[k]];

        sSqAvg1[[k]]=(sSqCum[[iLimit]]-
sSqCum[[1]])/nRegionPts;
        sSqAvg2[[k]]=(sSqCum[[jLimit]]-
sSqCum[[k]])/nRegionPts;

        dwrAvg1[[k]]=(dwrCum[[iLimit]]-
dwrCum[[1]])/wrSum1[[k]];
        dwrAvg2[[k]]=(dwrCum[[jLimit]]-
dwrCum[[k]])/wrSum2[[k]];

        dSqAvg1[[k]]=(dwrCum[[iLimit]]-
dwrCum[[1]])/wrSum1[[k]];
        dSqAvg2[[k]]=(dwrCum[[jLimit]]-
dwrCum[[k]])/wrSum2[[k]];

        (*dSqAvg1[[k]]=(dSqCum[[iLimit]]-
dSqCum[[1]])/nRegionPts;

```

```

        dSqAvg2[[k]]=(dSqCum[[jLimit]]-
dSqCum[[k]])/nRegionPts;*)

        sCbAvg1[[k]]=(sCbCum[[iLimit]]-
sCbCum[[1]])/nRegionPts;
        sCbAvg2[[k]]=(sCbCum[[jLimit]]-
sCbCum[[k]])/nRegionPts;

        rwrAvg1[[k]]=(rwrCum[[iLimit]]-
rwrCum[[1]])/wrSum1[[k]];
        rwrAvg2[[k]]=(rwrCum[[jLimit]]-
rwrCum[[k]])/wrSum2[[k]];

        rSqwrAvg1[[k]]=(rSqwrCum[[iLimit]]-
rSqwrCum[[1]])/wrSum1[[k]];
        rSqwrAvg2[[k]]=(rSqwrCum[[jLimit]]-
rSqwrCum[[k]])/wrSum2[[k]];

        rCbwrAvg1[[k]]=(rCbwrCum[[iLimit]]-
rCbwrCum[[1]])/wrSum1[[k]];
        rCbwrAvg2[[k]]=(rCbwrCum[[jLimit]]-
rCbwrCum[[k]])/wrSum2[[k]]
    ]
    ,{k,1,nEff} (* was kMin*)
];

(*****
**)
(* get overall averages
*)

(*****
**)
If[iOrder>=2,
  sAvg=sAvg1[[1]];
  sSqAvg=sSqAvg1[[1]];
  sCbAvg=sCbAvg1[[1]];

  rwrAvg=rwrAvg1[[1]];
  rSqwrAvg=rSqwrAvg1[[1]];
  rCbwrAvg=rCbwrAvg1[[1]];

];

e2rw=rSqwrAvg1[[1]]-rwrAvg1[[1]]2
];(*end regionAverages*)

```

```

(*****
***)
(* calc r1sAvg2
*)
(* these correlations are now AVERAGES
*)

(*****
***)
correlations=AbsoluteTiming[

    kList=Table[nEff-k+1,{k,1,nEff}];
    oneList=Table[1,{k,1,nEff}];

    (*ClearAll[gssFluct,grrFluct,gsrFluct,grsFluct];*)
    If [iOrder>=2,
        gww=ListCorrelate[wr, wr,1,0];

        sDiff=s;
        sDiffSq=sDiff2;

        dDiff=d;
        dDiffSq=dDiff2;

        (*nRTerms=4;
        ClearAll[x];
        rModel=Table[xi, {i,0,nRTerms-1}];
        rBkgFunct=
GeneralizedLinearModelFit[r[[1;;nEff]],rModel, x,Weights
□wr];
        rBkg=Table[rBkgFunct[i], {i,1,nEff}];
        rDiff=r-rBkg;*)
        rDiff=r;
        rDiffwr=rDiff*wr;
        rDiffSqwr=rDiffwr*rDiff;

        gss=ListCorrelate[swr,swr,1,0]/gww;
        gssFluct=gss-swrAvg1[[1]]*swrAvg2[[1]];
        gssFluct12=gss-swrAvg1*swrAvg2;
        gss1=ListCorrelate[(s-swrAvg1)*wr,(s-
swrAvg2)*wr,1,0]/gww;
        gssn=ListCorrelate[s,s,1,0]/kList;
        gsst=ListCorrelate[s,s,1,0];

        gdd=ListCorrelate[dwr,dwr,1,0]/gww;

```

```

gddFluct=gdd-dwrAvg1[[1]]*dwrAvg2[[1]];
gdFluct12=gdd-dwrAvg1*dwrAvg2;
gdd1=ListCorrelate[(d-dwrAvg1)*wr,(d-
dwrAvg2)*wr,1,0]/gww;
gddn=ListCorrelate[d,d,1,0]/kList;
gddt=ListCorrelate[d,d,1,0];

grr=ListCorrelate[rwr,rwr,1,0]/gww;
grrFluct=grr-rwrAvg1[[1]]^2;
grrFluct12=grr-rwrAvg1*rwrAvg2;
grr1=ListCorrelate[(r-rwrAvg1)*wr,(r-
rwrAvg2)*wr,1,0]/gww;
grrt=ListCorrelate[(r-rwrAvg1)*wr,(r-
rwrAvg2)*wr,1,0]*kList/gww;

gsr=ListCorrelate[swr,rwr,1,0]/gww;
gsrFluct12=gsr-swrAvg1*rwrAvg2;
gsrFluct      =gsr-swrAvg1[[1]]*rwrAvg2[[1]];
gsr1=ListCorrelate[(s-swrAvg1)*wr,(r-
rwrAvg2)*wr,1,0]/gww;

grs=ListCorrelate[rwr,swr,1,0]/gww;
grsFluct12=grs-rwrAvg1*swrAvg2;
grsFluct      =grs-rwrAvg1[[1]]*swrAvg2[[1]];
grs1=ListCorrelate[(r-rwrAvg1)*wr,(s-
swrAvg2)*wr,1,0]/gww;

gthth=ListCorrelate[th,th,1,0];
gth1=ListCorrelate[th^2,oneList,1,0];
gth2=ListCorrelate[oneList,th^2,1,0]
]
];(*end correlations*)

msd=(gth1+gth2-2*gthth)/kList;

(***** prob the neg pts are really zero
*****)
(* assume true val of -'ve pt actually zero *)
(* hc using implied g-factor *)
mlpNegVSum=Sum[If [vc[[i]]<0,0*(1/2)Log[2 π]+(1/2)Log[
n2vc]+vc[[i]]^2/(2 n2vc),0],{i,1,nEff}]; (* only neg pts are
suspect *)
mlpNegHSum=Sum[If [hc[[i]]<0,0*(1/2)Log[2 π ]+(1/2)Log[
n2hc]+hc[[i]]^2/(2 n2hc),0],{i,1,nEff}];
mlpNeg=mlpNegVSum+mlpNegHSum; (* vc,hc must both be
non-neg *)

```

```

(***** prob that DAvg really zero *****)
(* since 4>>1 bkg traces, ignore bkg variance *)
       $\sqrt{\text{variance [d]} / \text{nEff}}$ 
dAvgErr=
mlpDAvg=0*(1/2)Log[2 π]+1/2 (dAvg /dAvgErr)2; (* d=(vc-
bvc)-(hc-bhc) OK!!!! *)

(***** prob that DIntercept really zero *****)
sumy=Total[d];sumyy=Total[dSq];sumxy=Total[d*s];
sumx=Total[s];sumxx=Total[sSq];
denom=nEff*sumxx-sumx2;
intercept2P=(sumy*sumxx-sumxy*sumx)/denom;
slope2P=(nEff*sumxy-sumx*sumy)/denom;
var2P=(sumyy-intercept2P*sumy-slope2P*sumxy)/nEff;
interceptErr2P=
 $\sqrt{\text{var2P} * \text{sumxx} / \text{denom}}$ 
mlpDInt=0*(1/2)Log[2 π ]+1/2 (intercept2P
/interceptErr2P)2;

(***** prob that DSlope2P really zero *****)
slopeErr2P=
 $\sqrt{\text{var2P} * \text{nEff} / \text{denom}}$ 
mlpDSlope2P=0*(1/2)Log[2 π ]+1/2 (slope2P /slopeErr2P)2;

(***** prob that DSlope1P really zero *****)
slope1P=sumxy/sumxx;
var1P=(sumyy-slope1P*sumxy)/nEff;
slopeErr1P=
 $\sqrt{\text{var1P} * \text{nEff} / \text{denom}}$ 
mlpDSlope1P=0*(1/2)Log[2 π ]+1/2 (slope1P /slopeErr1P)2;

(***** prob that DFit really zero *****)
slope1P=sumxy/sumxx;
var1P=(sumyy-slope1P*sumxy)/nEff;
tFit=(dAvg-slope1P*sAvg)/
 $\sqrt{\text{var1P} / \text{nEff}}$ 
mlpFit=0*(1/2)Log[2 π ]+1/2 tFit2; (* d=(vc-bvc)-(hc-
bhc) OK!!!! *)

(***** prob that s,d uncorrelated *****)
rho=Correlation[s,d];
mlpDCorr=0*(1/2)Log[2 π ]+1/2 rho2 / (1-rho2) nEff;

mlpTot=mlpNeg;
mlpTot=mlpTot+mlpDCorr;
mlpTot=mlpTot+mlpDAvg;
mlpTot=mlpTot+mlpDInt;
mlpTot=mlpTot+mlpDSlope2P;
mlpTot=mlpTot+mlpDSlope1P;

```

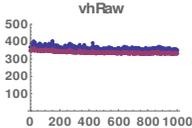
```

    mlpTot=mlpTot+ mlpFit;
    Return[mlpTot]
  );
  (*****
  ***)
  (* end of "depend" code
  *)
  (*****
  ***)

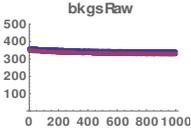
  (*v,h,iWt,iFirst,iLast,kFirst,kLast,nBkgs,
  g,c,b,f,aIn,
  iThreshMode,Thresh,iInit,iOrder*)

  depend[v,h,0,1,1000,0,0,4,0.40,0.,0.,0.,0.,0,3.,1,2]//
  AbsoluteTiming
  inFileName= 1000

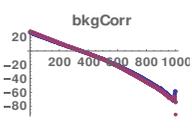
```



vhRaw



bkgsRaw



bkgCorr

```

  {
    ,
  }
  {0.091002,13.5331}

  (*****
  ***)
  (* process varying numbers of pts
  *)
  (*****
  ***)
  (*
  v,h,iWt,iFirst,iLast,kFirst,kLast,nBkgs,g,c,b,f,aIn,iThresh
  Mode,Thresh,iInit,iOrder
  *)
  depend[v,h,0,1,100,1,0,4,-
  0.40,0.0,0.0,0.0,0.0,0,3.0,1,2]// AbsoluteTiming
  {0.,depend[v,h,0,1,100,1,0,4,-0.4,0.,0.,0.,0.,0,3.,1,2]}
  {0.078000,10358.3}
  depend[v,h,0,1,1000,0,0,4,-
  0.40,0.0,0.0,0.0,0.0,0,3.0,1,2]// AbsoluteTiming
  {0.265201,1699.86}

  depend[v,h,0,1,1000,0,0,4,+0.40,0.0,0.0,0.0,0.0,0,3.0,1,2]//
  /AbsoluteTiming
  {0.249601,18778.9}
  depend[v,h,0,1,10000,0,0,4,-

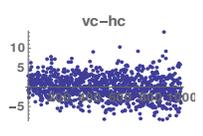
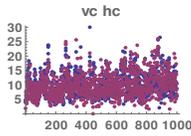
```



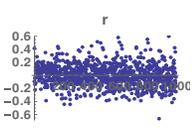
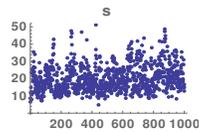
```

showOptResults[]
fitAnisDecay[]
fitIntenDecay[]
saveResults[]
savePlots[]
pInit= 271.514    g0= 0.3    c0= 0.    b0= 0.    f0= 0.
iPass= 1910
pMin= 3.61439*10-7    gMin= 0.389915    cMin= -2.5127    bMin= -
2.45579    fMin= 0.
Optimization NOT completed!
iPass=1910
neg=0    DAvG=2.98805e-7    DInt=1.83856e-8    DSlo2P=3.56529e-9
DSlo1P=3.48633e-8    fit=2.25472e-9    DCorr=3.56529e-9
p=3.61439e-7    g=0.389915    c=-2.5127    b=-2.45579    f=0

```



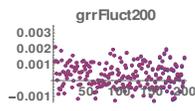
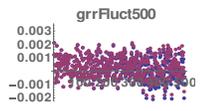
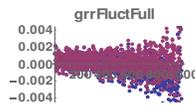
{ , }



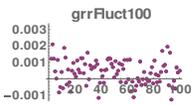
```

{ , }
pOpt= 3.61439*10-7    gNew= 0.389915    cNew= -2.5127    bNew=
-2.45579    fNew= 0.

```



{ , , , }



}

```

FIT RESULTS
params= {g0$328236->0.00249034,g∞$328236->
0.000251663,gTau$328236->2.40416}
errs= {2.85593*10-6,6.44024*10-8,0.00353535}
estdSD= 0.000895542
nlm[0]= 0.00249034

```

```

FIT RESULTS
params= {g0$331394->0.0025177,g∞$331394-
>0.000312165,gTau$331394->2.2368}
errs= {2.60529*10-6,6.32661*10-8,0.00299363}
estdSD= 0.000757096
nlm[0]= 0.0025177

```

```

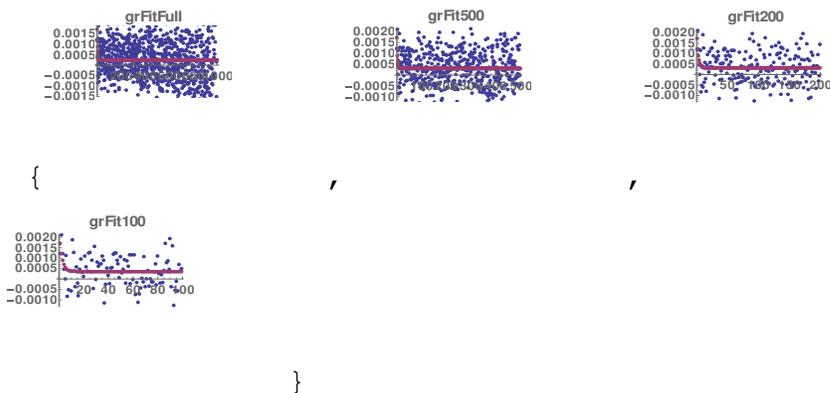
FIT RESULTS
params= {g0$332931->0.00252162,g∞$332931-
>0.000325196,gTau$332931->2.20465}
errs= {2.52635*10-6,8.8048*10-8,0.00287654}
estdSD= 0.000721435
nlm[0]= 0.00252162

```

```

FIT RESULTS
params= {g0$334168->0.00252948,g∞$334168-
>0.000362315,gTau$334168->2.11908}
errs= {2.57147*10-6,1.20166*10-7,0.00284529}
estdSD= 0.000700752
nlm[0]= 0.00252948

```



```

FIT RESULTS
params= {g0$335309->194.526,g∞$335309-
>2.51739,gTau$335309->3.2067}
errs= {0.0390577,0.00115967,0.00079947}
estdSD= 16.0735
nlm[0]= 194.526

```

```

FIT RESULTS
params= {g0$338466->194.789,g∞$338466-
>2.59879,gTau$338466->3.19736}
errs= {0.0309347,0.00106616,0.000631573}
estdSD= 12.6917
nlm[0]= 194.789

```



```

(*****
*****
(* maximize prob wrt g,b,c
*)
(*****
*****
ClearAll[optimize ];
optimize[]:=(
  (* ClearAll[g,c,b];*)
  ClearAll[g0,c0,b0,r,s,d];

  g0=.30;c0=0.0;b0=0.0;f0=0.0;
  (*g0=-.35 (*was-.4*);c0=320.0;b0=50.0;f0=1.0;*)

iWt=0;iFirst=1;iLast=1000;kFirst=0;kLast=0;nBkgs=4;alpha=0.
0;iThreshMode=0;thresh=0.;
  iOrderSave=4;

  progressFileName=base<>"-"<>"progress"<>"-
"<>DateString[]<>".csv";
  (*If [FileExistsQ[progressFileName],
  Close[progressFileName];
  DeleteFile[progressFileName]];
  prog=OpenWrite [progressFileName]; *)

  pInit=depend[v,h,iWt,iFirst,iLast,kFirst,kLast,nBkgs,g0,
c0, b0,f0,alpha,iThreshMode,thresh,1,iOrderSave];
  Print["pInit=",pInit," g0=",g0," c0=",c0, " b0=",
b0," f0=", f0];

  pMin=pInit;
  iPass=0;
  q={0,{{0,0},{0,0},{0,0},{0,0}}};
  Off[StringJoin::string];
  q=Monitor[

FindMinimum[{depend[v,h,iWt,iFirst,iLast,kFirst,kLast,nBkgs
,g,c,b,f,alpha,iThreshMode,thresh,1,0],(*b≥ 0&&c≥ 0&&*) -
1<g<1(*&& -1<f<1*)},{{g,g0},{b,b0},{c,c0}(*,{f,f0}*)},
  EvaluationMonitor:>
  (
    p=mlpTot;
    iPass=iPass+1;
    (*WriteString[prog,
gF[iPass],",",gF[p],",",gF[mlpNeg],",",gF[mlpBkgs],",",gF[mlpRAvg],",",gF[mlpDAvg],",",gF[mlpRhoSR],",",gF[mlpRhoRS],",
,"gF[mlpRhoSSR],",",gF[mlpRhoSRR],",",gF[g],",",gF[c],",",g

```

```

F[b], "\n"];*)
    If[p<pMin,
        gMin=g;cMin=c;bMin=b;pMin=p;fMin=f];

q[[2]][[1]][[2]]=g;q[[2]][[2]][[2]]=c;q[[2]][[3]][[2]]=b;q[
[2]][[4]][[2]]=f;
    If
[CurrentValue["ControlKey"]==True,Goto[continue]]
    ), (* end EvalMon*)
    PrecisionGoal->Automatic,
    AccuracyGoal -> Automatic,
    MaxIterations->10000(*Infinity*),
    WorkingPrecision->MachinePrecision,
    Method->Automatic
], (* end FindMin*)
"iPass="<>ToString[gF[iPass]]
<>" \np="<>ToString[gF[p]]
<>" neg="<> ToString[gF[mlpNeg]]
<>" DAvg="<> ToString[gF[mlpDAvg]]
<>" DInt2P="<> ToString[gF[mlpDInt]]
<>" DSlo2P="<> ToString[gF[mlpDSlope2P]]
<>" DSlo1P="<> ToString[gF[mlpDSlope1P]]
<> " fit="<>ToString[gF[mlpFit]]
<> " DCorr="<>ToString[gF[mlpDCorr]]
<>" \ng="<>ToString[gF[g]]
<>" c="<>ToString[gF[c]]
<> " b="<>ToString[gF[b]]
<> " f="<>ToString[gF[f]]
]; (*end Monitor*)

Label[continue ];
Print["iPass=",iPass," \npMin=",pMin," gMin=",gMin,"
cMin=", cMin, " bMin=",bMin, " fMin=",fMin];
If[q[[1]]==0 ,
    Print["Optimization NOT completed!"];

q[[2]][[1]][[2]]=gMin;q[[2]][[2]][[2]]=cMin;q[[2]][[3]][[2]]
]=bMin;q[[2]][[4]][[2]]=fMin,
    (*otherwise*)
    Print["Optimization successful!"]
];
(*Close[prog];*)

p=depend[v,h,iWt,iFirst,iLast,kFirst,kLast,nBkgs,gMin,cMin,
bMin,fMin,alpha,iThreshMode,thresh,1,2];
Print["iPass="<>ToString[gF[iPass]]
<>" \nneg="<> ToString[gF[mlpNeg]]

```

```

<>" DAVg="<> ToString[gF[mlpDAvg]]
<>" DInt="<> ToString[gF[mlpDInt]]
<>" DSlo2P="<> ToString[gF[mlpDSlope2P]]
<>" DSlo1P="<> ToString[gF[mlpDSlope1P]]
<> " fit="<>ToString[gF[mlpFit]]
<> " DCorr="<>ToString[gF[mlpDCorr]]
<>" \np="<>ToString[gF[p]]
<>" g="<>ToString[gF[gMin]]
<>" c="<>ToString[gF[cMin]]
<> " b="<>ToString[gF[bMin]]
<> " f="<>ToString[gF[fMin]]];
Print[optPlot={vchcPlot=ListPlot[{vc, hc}, PlotLabel->"vc
hc"], vchcDiffPlot=ListPlot[{vc-hc}, PlotLabel->"vc-hc"]]];
Print [{"\n", srPlot={sPlot=ListPlot[{s}, PlotLabel-
>"s"], rPlot=ListPlot[{r}, PlotLabel->"r"]}]}
);
optimize[];

```

```

pInit= 833.806    g0= 0.3    c0= 0.    b0= 0.    f0= 0.
iPass= 833
pMin= 830.496    gMin= 0.3    cMin= -0.00839591    bMin=
5.25733*10-9    fMin= 0.
Optimization NOT completed!
iPass=833
neg=244.842 DAVg=99.4498 DInt=0.250408 DSlo2P=164.877
DSlo1P=156.068 fit=0.131583 DCorr=164.877
p=830.496 g=0.3 c=-0.00839591 b=5.25733e-9 f=0

```



```
{ , }
```



```
{ , }
```

```

(*****
*****)
(*****
*****)
(*****
*****)

```



```

showOptResults[]:= (
  (*gNew=q[[2]][[1]][[2]];
  cNew=q[[2]][[2]][[2]];
  bNew=q[[2]][[3]][[2]];
  fNew=q[[2]][[4]][[2]];*)
  gNew=gMin;
  cNew=cMin;
  bNew=bMin;
  fNew=fMin;
  (*fNew=f0;*)
  (*gNew, cNew, bNew*)

Print["pOpt=", (*pOpt=*) depend[v,h,iWt,iFirst,iLast,kFirst,k
Last,nBkgs,gNew,cNew,bNew,fNew,0.,iThreshMode,
thresh,iInit, iOrder]," gNew=",gNew," cNew=",cNew,"
bNew=", bNew," fNew=",fNew]

  SetOptions[{Plot,ListPlot},ImageSize->{144,108}];
  ListPlot[{vc,hc},PlotRange-
>Automatic(*{0,100}*),PlotLabel->"vc,hc"];

gsrPlots={gsrFullPlot=ListPlot[{gsrFluct,gsrFluct12},PlotLa
bel->"gsrFluctFull"],

gsr500Plot=ListPlot[{gsrFluct[[;;500]],gsrFluct12[[;;500]]}
,PlotLabel->"gsrFluct500"],

gsr200Plot=ListPlot[{gsrFluct[[;;200]],gsrFluct12[[;;200]]}
,PlotLabel->"gsrFluct200"],

gsr100Plot=ListPlot[{gsrFluct[[;;100]],gsrFluct12[[;;100]]}
,PlotLabel->"gsrFluct100"]};

grsPlots={grsFullPlot=ListPlot[{grsFluct,grsFluct12},PlotLa
bel->"grsFluctFull"],

grs500Plot=ListPlot[{grsFluct[[;;500]],grsFluct12[[;;500]]}
,PlotLabel->"grsFluct500"],

grs200Plot=ListPlot[{grsFluct[[;;200]],grsFluct12[[;;200]]}
,PlotLabel->"grsFluct200"],

grs100Plot=ListPlot[{grsFluct[[;;100]],grsFluct12[[;;100]]}
,PlotLabel->"grsFluct100"]};

```

```

gssPlots={gssFullPlot=ListPlot[{gssFluct,gssFluct12},PlotLabel->"gssFluctFull"],

gss500Plot=ListPlot[{gssFluct[[;;500]],gssFluct12[[;;500]]},PlotLabel->"gssFluct500"],

gss200Plot=ListPlot[{gssFluct[[;;200]],gssFluct12[[;;200]]},PlotLabel->"gssFluct200"],

gss100Plot=ListPlot[{gssFluct[[;;100]],gssFluct12[[;;100]]},PlotLabel->"gssFluct100"]};

grrPlots={grrFullPlot=ListPlot[{grrFluct,grrFluct12},PlotLabel->"grrFluctFull"],

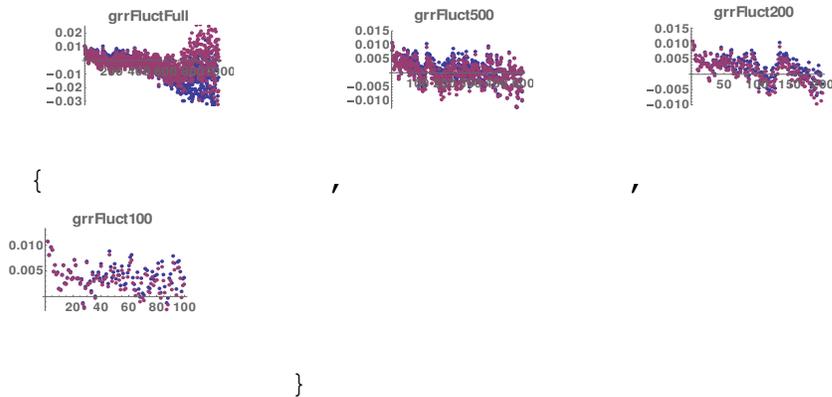
grr500Plot=ListPlot[{grrFluct[[;;500]],grrFluct12[[;;500]]},PlotLabel->"grrFluct500"],

grr200Plot=ListPlot[{grrFluct[[;;200]],grrFluct12[[;;200]]},PlotLabel->"grrFluct200"],

grr100Plot=ListPlot[{grrFluct[[;;100]],grrFluct12[[;;100]]},PlotLabel->"grrFluct100"]});
showOptResults[]

```

pOpt= 830.496 gNew= 0.3 cNew= -0.00839591 bNew= 5.25733*10⁻⁹ fNew= 0.



```

(*****
*****)
(*ANIS:  fit anis decays of various ranges
*)
(*****
*****)

```

```

ClearAll[fitAnisDecay];
fitAnisDecay[]:= (
  grFitData=grrFluct12;
  grFitDataName="grrFluct";
  ClearAll[grFitPlots,grFitPlotFull,grFitPlot500,
grFitPlot200, grFitPlot100];

  grFitFull=fitExpDecay[grFitData,
grFitDataName,gww,nEff,2,Length[grFitData]-1,"grFitFull"]
  grkFirstFull=2;grkLastFull=Length[grFitData];

grSDFull=gSD;grInfFull=gInf;grZeroFull=gZero;grTdFull=gTd;

grInfErrFull=gInfErr;grZeroErrFull=gZeroErr;grTdErrFull=gTd
Err;
  grFluctCalcFull=fluctCalc;grFitPlotFull=fitPlot;

  grFit500=fitExpDecay[grFitData,
grFitDataName,gww,nEff,2,500,"grFit500"]
  grkFirst500=2;grkLast500=500;
  grSD500=gSD;grInf500=gInf;grZero500=gZero;grTd500=gTd;

grInfErr500=gInfErr;grZeroErr500=gZeroErr;grTdErr500=gTdErr
;
  grFluctCalc500=fluctCalc;grFitPlot500=fitPlot;

  grFit200=fitExpDecay[grFitData,
grFitDataName,gww,nEff,2,200,"grFit200"]
  grkFirst200=2;grkLast200=200;
  grSD200=gSD;grInf200=gInf;grZero200=gZero;grTd200=gTd;

grInfErr200=gInfErr;grZeroErr200=gZeroErr;grTdErr200=gTdErr
;
  grFluctCalc200=fluctCalc;grFitPlot200=fitPlot;

  grFit100=fitExpDecay[grFitData,
grFitDataName,gww,nEff,2,100,"grFit100"]
  grkFirst100=2;grkLast100=100;
  grSD100=gSD;grInf100=gInf;grZero100=gZero;grTd100=gTd;

grInfErr100=gInfErr;grZeroErr100=gZeroErr;grTdErr100=gTdErr
;
  grFluctCalc100=fluctCalc;(grFitPlot100=fitPlot);

  Print[grFitPlots={grFitPlotFull,grFitPlot500,
grFitPlot200, grFitPlot100}]);
fitAnisDecay[]

```

```

FIT RESULTS
params= {g0$17385->0.00431244,g∞$17385->-
0.0048433,gTau$17385->279.502}
errs= {1.96231*10-6,3.39432*10-6,0.25542}
estdSD= 0.00404109
nlm[0]= 0.00431244

```

```

FIT RESULTS
params= {g0$20551->0.00634675,g∞$20551->-
0.00129116,gTau$20551->74.4138}
errs= {2.44084*10-6,9.53381*10-7,0.0474367}
estdSD= 0.00318781
nlm[0]= 0.00634675

```

NonlinearModelFit::sszero: The step size in the search has become less than the tolerance prescribed by the PrecisionGoal option, but the gradient is larger than the tolerance specified by the AccuracyGoal option. There is a possibility that the method has stalled at a point that is not a local minimum. >>

```

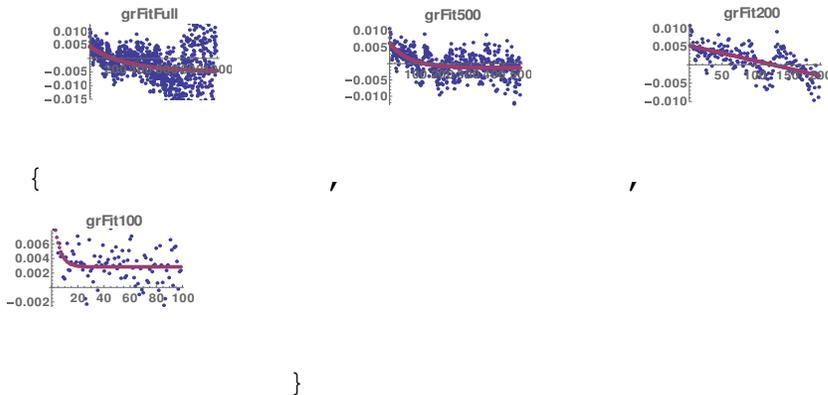
FIT RESULTS
params= {g0$22088->0.00524667,g∞$22088->-
176337.,gTau$22088->4.29153*109}
errs= {1.33572*10-6,52.1321,0.00214208}
estdSD= 0.00289532
nlm[0]= 0.00524667

```

```

FIT RESULTS
params= {g0$23813->0.0127492,g∞$23813-
>0.00285702,gTau$23813->4.5406}
errs= {7.07547*10-6,7.84871*10-7,0.00459721}
estdSD= 0.00214637
nlm[0]= 0.0127492

```



(*****)

```

*****
(*INTEN:  fit single exponential decay
*)
(*****
*****
)

ClearAll[fitIntenDecay];
fitIntenDecay[:]= (
  gsFitData=gssFluct12;
  gsFitDataName="gssFluct";

  gsFitFull=fitExpDecay[gsFitData,
gsFitDataName,gww,nEff,2,nEff,"gsFitFull"]
  gskFirstFull=2;gskLastFull=999;

gsSDFull=gSD;gsInfFull=gInf;gsZeroFull=gZero;gsTdFull=gTd;

gsInfErrFull=gInfErr;gsZeroErrFull=gZeroErr;gsTdErrFull=gTd
Err;
  gsFluctCalcFull=fluctCalc;gsFitPlotFull=fitPlot;

  gsFit500=fitExpDecay[gsFitData,
gsFitDataName,gww,nEff,2,500,"gsFit500"]
  gskFirst500=2;gskLast500=500;
  gsSD500=gSD;gsInf500=gInf;gsZero500=gZero;gsTd500=gTd;

gsInfErr500=gInfErr;gsZeroErr500=gZeroErr;gsTdErr500=gTdErr
;
  gsFluctCalc500=fluctCalc;gsFitPlot500=fitPlot;

  gsFit200=fitExpDecay[gsFitData,
gsFitDataName,gww,nEff,2,200,"gsFit200"]
  gskFirst200=2;gskLast200=200;
  gsSD200=gSD;gsInf200=gInf;gsZero200=gZero;gsTd200=gTd;

gsInfErr200=gInfErr;gsZeroErr200=gZeroErr;gsTdErr200=gTdErr
;
  gsFluctCalc200=fluctCalc;gsFitPlot200=fitPlot;

  gsFit100=fitExpDecay[gsFitData,
gsFitDataName,gww,nEff,2,100,"gsFit100"]
  gskFirst100=2;gskLast100=100;
  gsSD100=gSD;gsInf100=gInf;gsZero100=gZero;gsTd100=gTd;

gsInfErr100=gInfErr;gsZeroErr100=gZeroErr;gsTdErr100=gTdErr
;
  gsFluctCalc100=fluctCalc;(gsFitPlot100=fitPlot);

```

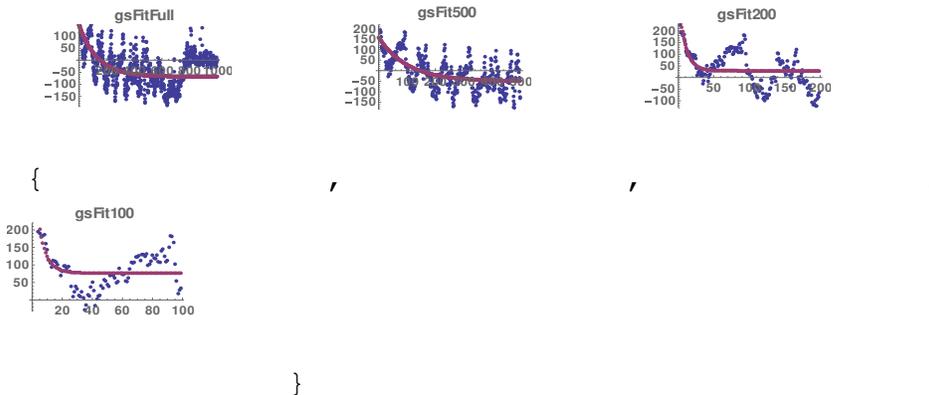
```
Print[gsFitPlots={gsFitPlotFull,gsFitPlot500,
gsFitPlot200, gsFitPlot100}}];
fitIntenDecay[];
```

```
FIT RESULTS
params= {g0$24960->147.946,g∞$24960->-68.1992,gTau$24960-
>133.181}
errs= {0.0382019,0.0223134,0.0551765}
estdSD= 63.2721
nlm[0]= 147.946
```

```
FIT RESULTS
params= {g0$28117->158.759,g∞$28117->-49.8709,gTau$28117-
>101.779}
errs= {0.0441285,0.0250898,0.0503983}
estdSD= 64.4431
nlm[0]= 158.759
```

```
FIT RESULTS
params= {g0$29654->346.354,g∞$29654->28.646,gTau$29654-
>9.58794}
errs= {0.134011,0.0179001,0.0061978}
estdSD= 65.5688
nlm[0]= 346.354
```

```
FIT RESULTS
params= {g0$30896->398.764,g∞$30896->76.7332,gTau$30896-
>5.27963}
errs= {0.129896,0.0163102,0.00310049}
estdSD= 43.7703
nlm[0]= 398.764
```



```
(*****
*****)
(* save results as CSV
```

```
*)
(*****
*****)
```

```
ClearAll[saveResults];
saveResults[:=(
  resultsFileName=base<>"-"<>"results"<>"-"
  "<>timeStampString<>".csv";
```

```

  If [FileExistsQ[resultsFileName] ,
    Close[resultsFileName];
    DeleteFile[resultsFileName]
  ];
  res=OpenWrite[resultsFileName];
  nameCol=1;
  valCol=2;
  ClearAll[rA]; (* rA= "Results Array" *)
  rA=Prepend[Table["",{i,nEff}], "param"];
  (*Print["Len=", Length[rA]]*)
  rA=appendCol[rA,Prepend[Table["",{i,nEff}], "value"]];
  rA=appendCol[rA,Prepend[Table[i,{i,nEff}], "iPt"]];
  rA=appendCol[rA,Prepend[veff, "veff"]];
  rA=appendCol[rA,Prepend[heff, "heff"]];
  rA=appendCol[rA,Prepend[bveff, "bveff"]];
  rA=appendCol[rA,Prepend[bheff, "bheff"]];
  rA=appendCol[rA,Prepend[vc, "vc"]];
  rA=appendCol[rA,Prepend[hc, "hc"]];
  rA=appendCol[rA,Prepend[s, "s"]];
  rA=appendCol[rA,Prepend[d, "d"]];
  rA=appendCol[rA,Prepend[r, "r"]];
  rA=appendCol[rA,Prepend[wr, "wr"]];
  rA=appendCol[rA,Prepend[gssFluct, "gssFluct"]];
  rA=appendCol[rA,Prepend[gssFluct12,
  "gssFluct12"]]; rA=appendCol[rA,Prepend[gsrFluct,
  "gsrFluct"]];
  rA=appendCol[rA,Prepend[gsrFluct12, "gsrFluct12"]];
  rA=appendCol[rA,Prepend[grsFluct, "grsFluct"]];
  rA=appendCol[rA,Prepend[grsFluct12, "grsFluct12"]];
  rA=appendCol[rA,Prepend[grrFluct, "grrFluct"]];
  rA=appendCol[rA,Prepend[grrFluct12, "grrFluct12"]];
  rA=appendCol[rA,Prepend[gww, "gww"]];
  rA=appendCol[rA,Prepend[grFluctCalcFull,
  "grFluctCalcFull"]];
  rA=appendCol[rA,Prepend[grFluctCalc500,
  "grFluctCalc500"]];
  rA=appendCol[rA,Prepend[grFluctCalc200,
  "grFluctCalc200"]];
```

```

    rA=appendCol[rA,Prepend[grFluctCalc100,
"grFluctCalc100"]] ;
    (*rA=appendCol[rA,Prepend[wTable, "grrFluctWts"]] ;*)
    (*rA//MatrixForm*)

    (*rA=addParam[rA,nameCol,valCol,"var1",1111];
rA=addParam[rA,nameCol,valCol,"var2",2222];*)
rA=addParam[rA,nameCol,valCol,"nb",NotebookFileName[]];
rA=addParam[rA,nameCol,valCol,"inFileName",inFileName];
rA=addParam[rA,nameCol,valCol,"iDot",iDot];

rA=addParam[rA,nameCol,valCol,"nEff",nEff];
rA=addParam[rA,nameCol,valCol,"cBkg",cBkg];
rA=addParam[rA,nameCol,valCol,"thMode",iThreshMode];
rA=addParam[rA,nameCol,valCol,"thLev",threshLevel];
rA=addParam[rA,nameCol,valCol,"nCorr",nCorr];
rA=addParam[rA,nameCol,valCol,"n2v",n2v];
rA=addParam[rA,nameCol,valCol,"n2h",n2h];
rA=addParam[rA,nameCol,valCol,"n2vc",n2vc];
rA=addParam[rA,nameCol,valCol,"n2sc",n2sc];

rA=addParam[rA,nameCol,valCol," "," "];

(*rA=addParam[rA,nameCol,valCol,"iOrder",iOrder];*)
rA=addParam[rA,nameCol,valCol,"kMin",kMin];
rA=addParam[rA,nameCol,valCol,"kMax",kMax];

rA=addParam[rA,nameCol,valCol,"mlpNeg",mlpNeg];
rA=addParam[rA,nameCol,valCol,"mlpDAvg",mlpDAvg];
rA=addParam[rA,nameCol,valCol,"mlpDInt",mlpDInt];

rA=addParam[rA,nameCol,valCol,"mlpDSlope2P",mlpDSlope2P];

rA=addParam[rA,nameCol,valCol,"mlpDSlope1P",mlpDSlope1P];
rA=addParam[rA,nameCol,valCol,"mlpFit",mlpFit];

rA=addParam[rA,nameCol,valCol,"iPass",iPass];
rA=addParam[rA,nameCol,valCol,"pMin",pMin];
rA=addParam[rA,nameCol,valCol,"gMin",gMin];
rA=addParam[rA,nameCol,valCol,"cMin",cMin];
rA=addParam[rA,nameCol,valCol,"bMin",bMin];
rA=addParam[rA,nameCol,valCol,"fMin",fMin];

rA=addParam[rA,nameCol,valCol," "," "];
rA=addParam[rA,nameCol,valCol,"rFitData",grFitDataName];

rA=addParam[rA,nameCol,valCol,"krFirstFull",grkFirstFull];r

```

```

A=addParam[rA,nameCol,valCol,"krLastFull",grkLastFull];
  rA=addParam[rA,nameCol,valCol,"grSDFull",grSDFull];

rA=addParam[rA,nameCol,valCol,"grZeroFull",grZeroFull];rA=ad
ddParam[rA,nameCol,valCol,"grZeroErrFull",grZeroErrFull];
  rA=addParam[rA,nameCol,valCol,"grInfFull",grInfFull];

rA=addParam[rA,nameCol,valCol,"grInfErrFull",grInfErrFull];
  rA=addParam[rA,nameCol,valCol,"grTdFull",grTdFull];

rA=addParam[rA,nameCol,valCol,"grTdErrFull",grTdErrFull];

  rA=addParam[rA,nameCol,valCol," "," "];
  rA=addParam[rA,nameCol,valCol,"rFitData",grFitDataName];

rA=addParam[rA,nameCol,valCol,"krFirst500",grkFirst500];rA=
addParam[rA,nameCol,valCol,"krLast500",grkLast500];
  rA=addParam[rA,nameCol,valCol,"grSD500",grSD500];

rA=addParam[rA,nameCol,valCol,"grZero500",grZero500];rA=add
Param[rA,nameCol,valCol,"grZeroErr500",grZeroErr500];rA=add
Param[rA,nameCol,valCol,"grInf500",grInf500];

rA=addParam[rA,nameCol,valCol,"grInfErr500",grInfErr500];
  rA=addParam[rA,nameCol,valCol,"grTd500",grTd500];
  rA=addParam[rA,nameCol,valCol,"grTdErr500",grTdErr500];

  rA=addParam[rA,nameCol,valCol," "," "];
  rA=addParam[rA,nameCol,valCol,"rFitData",grFitDataName];

rA=addParam[rA,nameCol,valCol,"krFirst200",grkFirst200];rA=
addParam[rA,nameCol,valCol,"krLast200",grkLast200];
  rA=addParam[rA,nameCol,valCol,"grSD200",grSD200];

rA=addParam[rA,nameCol,valCol,"grZero200",grZero200];rA=add
Param[rA,nameCol,valCol,"grZeroErr200",grZeroErr200];rA=add
Param[rA,nameCol,valCol,"grInf200",grInf200];

rA=addParam[rA,nameCol,valCol,"grInfErr200",grInfErr200];
  rA=addParam[rA,nameCol,valCol,"grTd200",grTd200];
  rA=addParam[rA,nameCol,valCol,"grTdErr200",grTdErr200];

  rA=addParam[rA,nameCol,valCol," "," "];
  rA=addParam[rA,nameCol,valCol,"rFitData",grFitDataName];

rA=addParam[rA,nameCol,valCol,"krFirst100",grkFirst100];rA=
addParam[rA,nameCol,valCol,"krLast100",grkLast100];

```

```

rA=addParam[rA,nameCol,valCol,"grSD100",grSD100];

rA=addParam[rA,nameCol,valCol,"grZero100",grZero100];rA=add
Param[rA,nameCol,valCol,"grZeroErr100",grZeroErr100];
rA=addParam[rA,nameCol,valCol,"grInf100",grInf100];

rA=addParam[rA,nameCol,valCol,"grInfErr100",grInfErr100];
rA=addParam[rA,nameCol,valCol,"grTd100",grTd100];
rA=addParam[rA,nameCol,valCol,"grTdErr100",grTdErr100];

rA=addParam[rA,nameCol,valCol," "," "];
rA=addParam[rA,nameCol,valCol,"sFitData",gsFitDataName];

rA=addParam[rA,nameCol,valCol,"ksFirstFull",gskFirstFull];
rA=addParam[rA,nameCol,valCol,"ksLastFull",gskLastFull];
rA=addParam[rA,nameCol,valCol,"gsSDFull",gsSDFull];
rA=addParam[rA,nameCol,valCol,"gsZeroFull",gsZeroFull];

rA=addParam[rA,nameCol,valCol,"gsZeroErrFull",gsZeroErrFull
];rA=addParam[rA,nameCol,valCol,"gsInfFull",gsInfFull];

rA=addParam[rA,nameCol,valCol,"gsInfErrFull",gsInfErrFull];
rA=addParam[rA,nameCol,valCol,"gsTdFull",gsTdFull];

rA=addParam[rA,nameCol,valCol,"gsTdErrFull",gsTdErrFull];

rA=addParam[rA,nameCol,valCol," "," "];
rA=addParam[rA,nameCol,valCol,"sFitData",gsFitDataName];
rA=addParam[rA,nameCol,valCol,"ksFirst500",gskFirst500];
rA=addParam[rA,nameCol,valCol,"ksLast500",gskLast500];
rA=addParam[rA,nameCol,valCol,"gsSD500",gsSD500];
rA=addParam[rA,nameCol,valCol,"gsZero500",gsZero500];

rA=addParam[rA,nameCol,valCol,"gsZeroErr500",gsZeroErr500];
rA=addParam[rA,nameCol,valCol,"gsInf500",gsInf500];

rA=addParam[rA,nameCol,valCol,"gsInfErr500",gsInfErr500];
rA=addParam[rA,nameCol,valCol,"gsTd500",gsTd500];
rA=addParam[rA,nameCol,valCol,"gsTdErr500",gsTdErr500];

rA=addParam[rA,nameCol,valCol," "," "];
rA=addParam[rA,nameCol,valCol,"sFitData",gsFitDataName];
rA=addParam[rA,nameCol,valCol,"ksFirst200",gskFirst200];
rA=addParam[rA,nameCol,valCol,"ksLast200",gskLast200];
rA=addParam[rA,nameCol,valCol,"gsSD200",gsSD200];
rA=addParam[rA,nameCol,valCol,"gsZero200",gsZero200];

```

```

rA=addParam[rA,nameCol,valCol,"gsZeroErr200",gsZeroErr200];
rA=addParam[rA,nameCol,valCol,"gsInf200",gsInf200];

rA=addParam[rA,nameCol,valCol,"gsInfErr200",gsInfErr200];
  rA=addParam[rA,nameCol,valCol,"gsTd200",gsTd200];
  rA=addParam[rA,nameCol,valCol,"gsTdErr200",gsTdErr200];

  rA=addParam[rA,nameCol,valCol," "," "];
  rA=addParam[rA,nameCol,valCol,"sFitData",gsFitDataName];
  rA=addParam[rA,nameCol,valCol,"ksFirst100",gskFirst100];
  rA=addParam[rA,nameCol,valCol,"ksLast100",gskLast100];
  rA=addParam[rA,nameCol,valCol,"gsSD100",gsSD100];
  rA=addParam[rA,nameCol,valCol,"gsZero100",gsZero100];

rA=addParam[rA,nameCol,valCol,"gsZeroErr100",gsZeroErr100];
rA=addParam[rA,nameCol,valCol,"gsInf100",gsInf100];

rA=addParam[rA,nameCol,valCol,"gsInfErr100",gsInfErr100];
  rA=addParam[rA,nameCol,valCol,"gsTd100",gsTd100];
  rA=addParam[rA,nameCol,valCol,"gsTdErr100",gsTdErr100];

  rA=addParam[rA,nameCol,valCol," "," "
"];rA=addParam[rA,nameCol,valCol,"grrFluct12[[2]]",grrFluct
12[[2]]];

rA=addParam[rA,nameCol,valCol,"grrFluct12[[3]]",grrFluct12[
[3]]];

  Do>(*i down*)
    jLast=Length[rA[[i]]];
    Do>(*j across*)
      WriteString[res,ToString[ff[rA[[i,j]]]]];

If[j!=jLast,WriteString[res,","],WriteString[res,"\n"]]
  ,{j,1,jLast}
]
  ,{i,Length[rA]}
];
Close[res]);
saveResults[]
2015-01-31-IgE-1-LRB-d01-results-20161108-150450.csv

(*****
*****
(* combine all graphics and export
*)
(*****

```

```

*****
ClearAll[savePlots];
savePlots[]:=
  (ClearAll[pageLabel];

  (*bs(*BaseStyle*)={FontFamily["CourierNew",FontSize["12"];*)
  pageLabel=
  Framed[
    Graphics[
      Inset[
        Pane[
          TextCell[
            StringJoin[inFileName," =
inFileName","\n",ToString[ iDot ] ," = iDot" ,
"\n",FileBaseName[NotebookFileName[]]," =
notebook","\n",dateString," = date","\n", timeString, " =
time"],
            "Text",
            FontSize->10
          ],
          144 (*pane width pts*)
        ],
        {0,0} (*inset pos*)
      ],
      ImageSize->{144,108}
    ]
  ];
  allPlots=GraphicsGrid[{
    {pageLabel,vhRawPlot,bkgRawPlot, bkgCorrPlot},
    {vchcPlot,vchcDiffPlot,sPlot,rPlot},
    {gsrFullPlot, gsr500Plot ,gsr200Plot, gsr100Plot},
    {gssFullPlot,gss500Plot,gss200Plot,gss100Plot},
    {grrFullPlot,grr500Plot,grr200Plot,grr100Plot},
    {gsFitPlotFull,gsFitPlot500, gsFitPlot200,
gsFitPlot100},
    {grFitPlotFull,grFitPlot500, grFitPlot200,
grFitPlot100}
  }];
  plotFileName=base<>"-"<>"plots"<>"-
"<>timeStampString<>".pdf";
  Export [plotFileName, allPlots]);
savePlots[]

2015-01-31-IgE-1-LRB-d01-plots-20161108-150450.pdf

(*****
***)

```

```

(* test execution time
*)
(*****
***)
nEff
createArrays[[1]]
fillArrays[[1]]
regionAvg[[1]]
correlations[[1]]
(*distCorr[[1]]*)

(*****
***)
(* attempt single probability calc'
*)
(* takes 28 mSec (iWt=0) or 37 mSec (iWt=1)
*)
(* graph data
*)
(* "mlp" means "Minus Log Probability"
*)
(*****
***)
ClearAll[]
g0=+.30; (*g0=-.364;*)
c0=320.0; (*c0=324.6;*)
b0=200.0;
(*b0=40.5;*)
f0=.35;

iWt=0;
iFirst=0;
iLast=0;
kFirst=0;
kLast=0;
nBkgs=4;
 $\alpha$ =0.0;
iThreshMode=0;
thresh=3.0;
iInit=1;

q=depend[v,h,iWt,iFirst,iLast,kFirst,kLast,nBkgs,
g0,c0,b0,f0, $\alpha$ ,iThreshMode,thresh,iInit,iOrder];
If[q<qMin,qMin=q];
If [(q==0 || q=="Indeterminate" || q<qOld),
Print["prob=",q, "* pPrev=", qOld, " pMin=",qMin, "
g0=",g0," c0=",c0, " b0=", b0],

```

```

Print["prob=",q, " pPrev=", qOld, " pMin=",qMin, "
g0=",g0," c0=",c0, " b0=", b0]
]
qOld=q;

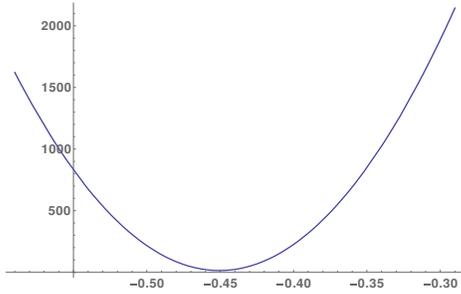
Print["varS=", Variance[s], " sAvg=",sAvg,"
mlpBkgs=",mlpBkgs]
Print["mlpNeg=",mlpNeg," mlpDAvg=",mlpDAvg,"
mlpBkgs=",mlpBkgs," mlpRho=", mlpRho, " mlpRhosr=",
mlpRhosr, " mlpRhosrr=", mlpRhosrr, " mlpRhosrr=",
mlpRhosrr];
Print["ρ11=",mlpRho11," ρ21=",mlpRho21," ρ12=",mlpRho12,"
ρ31=",mlpRho31," ρ22=",mlpRho22," ρ13=",mlpRho31];
ListPlot[{vc,hc, vc-hc}]

```

```

(*****
*****
(* estimate initial parameters
*)
(*****
*****)Plot[(*Log10[*]depend[v,h,0,1,1000,0,0,g0,0.0,0.0
,0.0(*α*),0(*iInit*),1.0(*f*),0(*iOrder*)](*)*),{g0,-0.59,-
0.29},(*PlotRange->{0,6},*)PlotPoints->4,ImageSize->Medium]

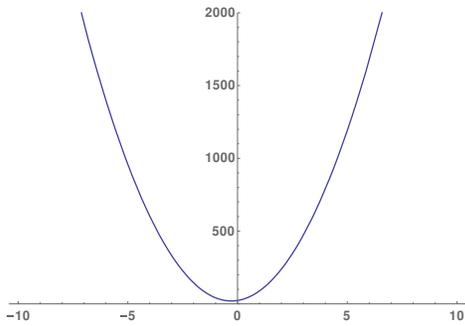
```



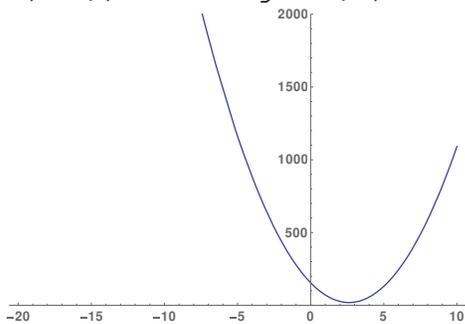
```

Plot[(*Log10[*]depend[v,h,0,1,1000,0,0,-
.44,c0,0.1,0.0(*α*),0(*iInit*),1.0(*f*),0(*iOrder*)](*)*),{
c0,-10,10},PlotRange->{0,2000},PlotPoints->4,ImageSize-
>Medium]

```

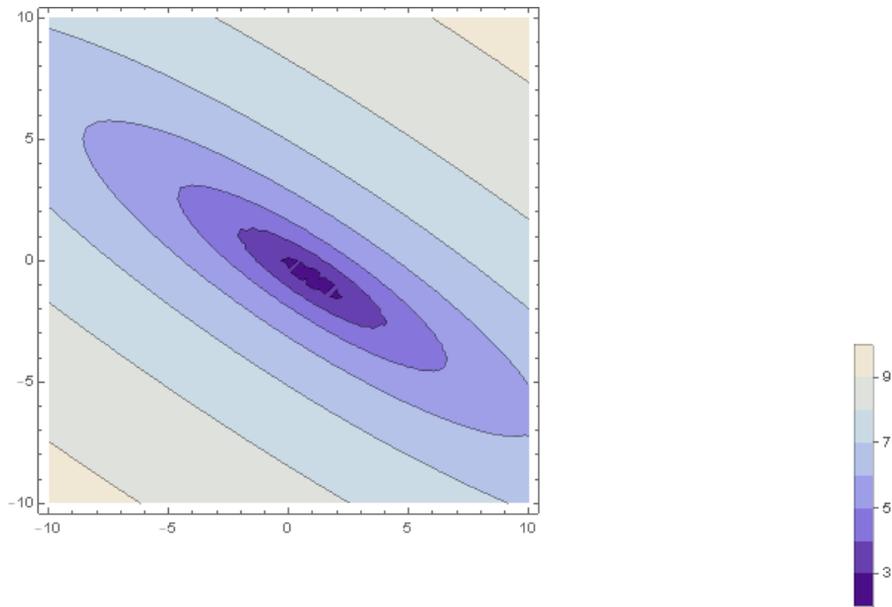


```
Plot[(*Log10[*]depend[v,h,0,1,1000,0,0,-.44,-
2.0,b0,0.0(*α*),0(*iInit*),1.0(*f*),0(*iOrder*)](*)*),{b0,-
20,10},PlotRange->{0,2000},PlotPoints->4,ImageSize->Medium]
```



```
ContourPlot[Log[depend[v,h,0,1,1000,0,0,-
0.45,c0,b0,0.0(*α*),0(*iInit*),0.0(*f*),0(*iOrder*)]],{c0,-
10.0,10.0},{b0,-10.0,10.0},PlotPoints->4]
$Aborted
```

```
ListContourPlot[ParallelTable[Log[depend[v,h,0,1,1000,0,0,-
0.445,c0,b0,0.0(*α*),0(*iInit*),1.0(*f*),0(*iOrder*)]],{c0,-
-10.0,10.0,0.5},{b0,-10.0,10.0,0.5}],DataRange->{{-
10,10},{-10,10}},PlotRange-
>Automatic(*{0,300}*),PlotLegends->Automatic]
{
```



```

{
,
}

(*****
*****)
(* calculate anisotropy and intensity fluctuations
*)
(*****
*****)
SetOptions[{Plot,ListPlot},ImageSize-
>{144,108}];gsrPlots={gsrFullPlot=ListPlot[{gsrFluct,gsrFlu
ct12},PlotLabel->"gsrFluctFull"],

gsr500Plot=ListPlot[{gsrFluct[[;;500]],gsrFluct12[[;;500]]}
,PlotLabel->"gsrFluct500"],

gsr200Plot=ListPlot[{gsrFluct[[;;200]],gsrFluct12[[;;200]]}
,PlotLabel->"gsrFluct200"],

gsr100Plot=ListPlot[{gsrFluct[[;;100]],gsrFluct12[[;;100]]}
,PlotLabel->"gsrFluct100"]}

gssPlots={gssFullPlot=ListPlot[{gssFluct,gssFluct12},PlotLa
bel->"gssFluctFull"],

gss500Plot=ListPlot[{gssFluct[[;;500]],gssFluct12[[;;500]]}

```

```
,PlotLabel->"gssFluct500"],  
  
gss200Plot=ListPlot[{gssFluct[[;;200]],gssFluct12[[;;200]]}  
,PlotLabel->"gssFluct200"],  
  
gss100Plot=ListPlot[{gssFluct[[;;100]],gssFluct12[[;;100]]}  
,PlotLabel->"gssFluct100"]}  
  
grrPlots={grrFullPlot=ListPlot[{grrFluct,grrFluct12},PlotLa  
bel->"grrFluctFull"],  
  
grr500Plot=ListPlot[{grrFluct[[;;500]],grrFluct12[[;;500]]}  
,PlotLabel->"grrFluct500"],  
  
grr200Plot=ListPlot[{grrFluct[[;;200]],grrFluct12[[;;200]]}  
,PlotLabel->"grrFluct200"],  
  
grr100Plot=ListPlot[{grrFluct[[;;100]],grrFluct12[[;;100]]}  
,PlotLabel->"grrFluct100"]}
```

Program Code 2: Image Corr v.122

```

(* GOAL: assign bkg and g-factor to QD traces *)
(* so as to minimize dependence of r and s *)
(* use the new defn of g as 1+f and 1-f *)
(* mpy 2 reals= $\sim 0.4\mu s$  div 2 reals= $\sim 8\mu s$  *)
(*****
)

(*****
)
(* initialize Mathematica system
*)
(*****
)
ClearSystemCache[];
SetDirectory[NotebookDirectory[]];
Needs["CCompilerDriver`"];
Needs["SymbolicC`"];
Needs["CompiledFunctionTools`"];
Unprotect[CompiledFunctionTools`Private`getInstruction];
CompiledFunctionTools`Private`getInstruction[line_,{0,_}]:=
CompiledFunctionTools`Private`getInstruction[line,{3,1}]
SetDirectory[NotebookDirectory[]];
(*SetOptions[$FrontEndSession,PrintingStyleEnvironment<"Working"];*)
Off[General::munfl];
{\pard{}}[FittedModel::precw];
Off[NonlinearModelFit::sszero];

ClearAll[gPrint,
one,remOdd,dP,aP,se,restIPrint,saveIPrint];
gPrint= If[iPrint!=0,Print[##]]&;
iPrint=1;
iPrintS=1;
one=#1&;
remOdd[e_, x_] := ((e/.{x->Power[x,2]})/.{ x2->0, x6->0, x10->0, x14-> 0})/.{x-> x1/2}
SetAttributes[dP, HoldAll];
dP[z_] := If[(iPrint!=0 && ValueQ[z]==True) ||
(ValueQ[iPrint]==False), Print[HoldForm[z], "=", ReleaseHold[z]
](*//MatrixForm*)]]; (* debug print showing variable name*)
aP[a_] := Module[{last}, last=Min[8,
Length[a]]; Evaluate[dP[a[[1;;last]]]]; (*array print
showing 8 list members max*)
se[x_] := Simplify[Expand[x]]; (* simplify-expand *)
restIPrint[] := (iPrint=iPrintS);

```

```

saveIPrint[]:=(iPrintS=iPrint);

(*****
**)
(*  global parameters
*)
(*****
**)
(*inFileName="IgE-1-LRB.csv";*)
iFirstDot=1; iLastDot=1; (*"0" means all dots*)
iDot=1; (* used when NOT entering via "dataImport"*)
iPrint=1;
cBkg=300.0(*319.75*); (* initial value of camera bkg *)
isStamp=True;
iWt=2;(*2=Gaussian; 1=Poisson*)
iGeom=1;(*1=polariz, 2=anis*)
iPrint=1; (* print all dP instances *)
pGoal=0.01;
maxPass=2000;
g0=0.4;b0=0.0;c0=0.0;f0=-1.; (*"-1" means use prog's camera
bkg *)
iFirst=2;iLast=1001; kFirst=0; kLast=0;
nBkgs=4;
isBkgs=1;
isSim=0; (* simulation? *)

(*****
*)
(*****
*)
(*  UTILITIES  *)
(*****
*)
(*****
*)

(*****
)
(*  set graphics point sizes (fract of total graph)
*)
(*****
)
SetOptions[{Plot,ListPlot, ListLinePlot},
  PlotStyle->PointSize[0.01],
  PlotStyle->{RGBColor[0.2472,0.24,0.6],
    RGBColor[0.6,0.24,0.4429],
    RGBColor[0.6,0.5470,0.24],

```

```

    RGBColor[0.24,0.6,0.3369]},
BaseStyle->{
  FontFamily->"Helvetica",
  FontWeight->"Bold" },
ImageSize->{144,108} (*Medium*)
];

(*****
*****
(*  define pattern replace to pull constants out of sums
*)
(*****
*****
(* patt /; test is a pattern which matches only if the
evaluation of test yields True *)
(* lhs  $\square$  rhs "\\\"\\\"\\\"RuleDelayed\\\"\\\"\\\"\\\" represents a
rule that transforms lhs to rhs evaluating rhs only after
the rule is used *)
(* FreeQ[expr,form] yields True if no subexpression in expr
matchesformand yields False *)
ClearAll[BringOut,outrules];
outrules={Sum[f_+
g_,it:{x_Symbol,__}]>Sum[f,it]+Sum[g,it],Sum[c_
f_,it:{x_Symbol,__}]>c
Sum[f,it]/;FreeQ[c,x],Sum[c_,it:{x_Symbol,__}]>c
Sum[1,it]/;FreeQ[c,x]];
BringOut[s_]:=s //. outrules
(*BringOut [Sum [c*i*x[[i]]1,{i,n}]];*

(*****
*****
(*  map Sum function over sum of terms
*)
(*****
*****
(*ClearAll[x,y,z,qd0,n];
qd0= x[[i]] +2y[[i]]^2+ 3 z[[i]]^3;
gd0=Total[Sum[#, {i,1,n}]&/@(List@@qd0)];
BringOut[gd0]*)

(*****
*****
(*  abbreviate FortranForm for printing
*)
(*****
*****
ClearAll[ff];

```

```

ff[x_]:=FortranForm[x];

(*****
*****
(*  define gRound to get suitable form printing large,
small nums  *)
(*****
*****
ClearAll[gR];
gR[x0_,sf_]:= (* "g Round" *)Module[{e,xr,x},
  (x=Re[x0];e=MantissaExponent[x][[2]]-sf;
  xr=(10^e)*Round[x*10^(-e)];
  xr)];

(*****
*****
(*  define gF "George Format" for compact 1-line output
*)
(*****
*****
ClearAll[gF];
gF[x_]:=ff[gR[x,6]];

(*****
*)
(*****
*)
(*  FUNCTIONS THIS PROGRAM  *)
(*****
*)
(*****
*)

(*****
)
(*          SZEKELY dist covariance via Hu's algorithm
*)
(*****
)
ClearAll[distCovCompile];
distCovCompile=Compile[
  {{xIn,_Real,1},
  {yIn,_Real,1}},

Module[{index,si,sLast,t,ax,v,nw,zeros,idx,iv1,iv2,iv3,iv4,
i,r,s,gap,k,idxr,csumv,idx1,idx2,st1,st2,e1,e2,kf,covterm,c

```

```

1, c2, c3, c4, d, ySorted, by, nsq, ncb, nq, term1, term2, term3, x, y, n,
j}},
  (*iPrint=0;*)
  n=Length[xIn];
  index=Ordering[xIn];
  x=Sort[xIn]; (*gPrint["xSorted=",x//MatrixForm];*)
  y=yIn[[index]]; (*gPrint["yReordered=",y//MatrixForm];*)
  si=Accumulate[x]; (*gPrint["si=",si//MatrixForm];*)
  sLast=si[[n]];
  t=Table[i, {i,-(n-
2), n, 2}]; (*gPrint["t=",t//MatrixForm];*)
  ax=t*x+(sLast-2 si); (*gPrint["ax=",ax//MatrixForm];*)
  v={x,y,x*y}□; (*gPrint["v=",v//MatrixForm];*)
  nw=Dimensions[v][[2]];
  zeros=Table[0,n];

idx={Table[i,{i,n}],zeros}□; (*gPrint["idx=",idx//MatrixForm
];*)
  iv1=Table[0.,n];iv2=iv1;iv3=iv1;iv4=iv1;
  i=1;r=1;s=2;

  (*While [i<n,*)
  Label[startWhileI];
  If[i>=n, Goto[endWhileI]];
  gap=2*i;
  k=0;
  idxr=idx[[All,r]]; (*gPrint["idxr=",idxr//MatrixForm];*)
  csumv=Prepend[Accumulate[v[[idxr]],Table[0,nw]];
  (* OK to here*)
  (*gPrint["csumv=", csumv//MatrixForm];*)

  (*For [j=1, j<n, j=j+gap,*)
  j=1;
  Label[startForJ];
  If [j>=n, Goto[endForJ]];
  (*st1=j;e1=Min[st1+i-1,n];st2=j+i;e2=Min[st2+i-1,n];*)
  st1=j;e1=st1+i-1; If[e1>=n,e1=n];st2=j+i;e2=st2+i-
1;If[e2>=n,e2=n];
  (*While [(st1≤e1)&&(st2≤e2),*)
  Label[startWhileSt];
  If[(st1>e1)|| (st2>e2),Goto[endWhileSt]];
  k=k+1;(*gPrint ["i=",i," j=",j, " k=",k, " st1=",st1,"
e1=",e1," st2=",st2," e2=",e2];*)
  idx1=idxr[[st1]];idx2=idxr[[st2]];
  (*gPrint ["idx1=",idx1," idx2=",idx2];*)
  If[ y[[idx1]]>=y[[idx2]],
  idx[[k,s]]=idx1;

```

```

    st1=st1+1,
    (*else*)
    idx[[k,s]]=idx2;
    st2=st2+1;
    iv1[[idx2]]      =iv1[[idx2]]+e1-st1+1;iv2[[idx2]]
=iv2[[idx2]]+(csumv[[e1+1,1]]-csumv[[st1,1]]);

    iv3[[idx2]]      =iv3[[idx2]]+(csumv[[e1+1,2]]-
csumv[[st1,2]]);
    iv4[[idx2]]      =iv4[[idx2]]+(csumv[[e1+1,3]]-
csumv[[st1,3]]);
    ];(*end If[y[idx1...]*)
    (*gPrint ["ivN=",{iv1,iv2,iv3,iv4}□//MatrixForm];*)

    (*); (*end While[(st1...]*)*)
    Goto[startWhileSt];
    Label[endWhileSt]; (*end While[(st1...]*)*)

    (*gPrint["i,j=",i," ",j];*)
    If[ st1<=e1,
    kf=k+e1-st1+1; (*gPrint["kf=",kf];*)
    idx[[k+1];;kf,s]=idxr[[st1;;e1]];
    k=kf,
    (*else*)
    If[st2<=e2,
    kf=k+e2-st2+1;
    (*gPrint[idx[[k+1];;kf,s]]);*)
    idx[[k+1];;kf,s]=idxr[[st2;;e2]];
    k=kf
    ];
    ];(*end If[st2...]*)

    (*gPrint["kf=",kf," idx=",idx//MatrixForm];*)
    (*gPrint["idx=",idx//MatrixForm];*)
    (*);(* end For[j=1...]*)*)
    j=j+gap;
    Goto[startForJ];
    Label[endForJ];(* end For[j=1...]*)

    i=gap;
    r=3-r;s=3-s;
    (*);(* end While i<n*)*)
    Goto[startWhileI];
    Label[endWhileI];(* end While i<n*)*)
    covterm=n*(x-Mean[x]).(y-Mean[y]);

    c1=iv1.v[[All,3]];

```

```

c2=Total[iv4];
c3=iv2.y;
c4=iv3.x;

d=4*((c1+c2)-(c3+c4))-2*covterm;
ySorted=y[[idx[[Table[i,{i,n,1,-1}],r]]]];
si=Accumulate[ySorted];
sLast=si[[n]];
by=Table[0.,n];
by[[idx[[n;;1;;-1,r]]]]=Table[i,{i,-(n-
2),n,2}]*ySorted+(sLast-
2*si);nsq=N[n*n];ncb=nsq*n;nq=ncb*n;term1=d/nsq;term2=2*(ax
.by)/ncb;term3=Total[ax]*Total[by]/nq;
Return[N[(term1+term3)-term2]]
],(*end module/function*)
RuntimeAttributes->{Listable},Parallelization->False,
CompilationTarget->"C",RuntimeOptions->"Speed"
];(*end compile*)
<<CompiledFunctionTools`
(*CompilePrint[distCovCompile]*)
(*yIn=N[{3,5,7,3,8,4,6,7}];Print["yIn=",yIn];
xIn=N[{1,5,3,2,4,6,7,5}];Print["xIn=",xIn];
Print["distCov=",distCovCompile[xIn,yIn]];*)

(*****
*)
(* SZEKELY dist correlation from dist covariances
*)
(*****
*)
ClearAll[distCorrFunct];
distCorrFunct[x_,y_]:=Module[{covAB,covAA,covBB},
covAB=Evaluate[distCovCompile[x,y]];
covAA=Evaluate[distCovCompile[x,x]];
covBB=Evaluate[distCovCompile[y,y]];

(*Print["covAB=",covAB,"\ncovAA=",covAA,"\ncolBB=",covBB];*
)
Return[N[covAB/Sqrt[covAA*covBB]]];
];
(*yIn=N[{3,5,7,3,8,4,6,7}];Print["yIn=",yIn];
xIn=N[{1,5,3,2,4,6,7,5}];Print["xIn=",xIn];
Print["distCorr=",distCorrFunct[xIn,yIn]];*)

(*****
*****
*)
(* fast autocorrelation using LISTCORRELATE

```

```

*)
(*****
*****
ClearAll[dataCorrelate];
dataCorrelate[t_,u_]:=Module[{n},
z=ListCorrelate[t,u,1,0];n=Length[t];Table[z[[i]]/(n-
i+1),{i,n}]];
(*t={4,3,2,1}
u={1,2,3,4}
z=ListCorrelate[t,u,1,0]
z1=Table[z[[i]]/(Length[z]-i+1),{i,Length[z]}]
dataCorrelate[t,u]*)

(*****
*****
(* weighted covariance function
*)
(*****
*****
ClearAll[wtdCov];
wtdCov[f_,g_,w_]:=Module[{n,wTot,temp},
n=Length[f];
wTot=Total[w];
temp=(wTot*Total[f*g*w]-Total[f*w]*Total[g*w])/wTot2
]; (* fails for f=g and i=1 *)

(*****
*****
(* weighted correlation coeff
*)
(*****
*****
ClearAll[wtdCorrCoeff];
wtdCorrCoeff[f_,g_,w_]:=Module[{} ,
wtdCov[f,g,w]/\[Sqrt](wtdCov[f,f,w]*wtdCov[g,g,w])];

(*****
)
(* faster version of brute force correlate
*)
(*****
)
SetSystemOptions["CompileOptions"-
>{"CompileReportExternal"->True}];
SetSystemOptions["CompileOptions"-
>{"CompileReportExternal"->False}];
ClearAll[trueCorr,i];

```

```

trueCorr=Compile[(* wx, wy are just 1/vx, 1/vy for each pt
*)

{{x,_Real,1},{y,_Real,1},{wx,_Real,1},{wy,_Real,1}},Module[
{n,nFact,tau,txy=0.0,tx=0.0,ty=0.0,
tw=0.0,w=0.0,iy,ix,vx=0.0,vy=0.0,xVar,yVar,
x0=0.0,y0=0.0,kList,gxy,gx,gy,gw},
n=Length[x];
gxy=Table[0.0,{n}];gx=gxy;gy=gxy;gw=gxy;
xVar=1.0/wx;yVar=1.0/wy;
kList=Table[n-tau+1,{tau,n}];
For[tau=1,tau<=n,tau++,
txy=0.0;tx=0.0;ty=0.0;tw=0.0;
For[ix=1,ix<=n-tau+1,ix++,
iy=ix+tau-1;
x0=x[[ix]];y0=y[[iy]];
vx=xVar[[ix]];vy=yVar[[iy]];
w=(vy*x0^2+vx y0^2+vx*vy)^-1;
txy+=w*x0*y0;tx += w*x0;ty+=w*y0;tw+=w
];(*for i*)
gxy[[tau]]=txy; (* weighted sums, not averages*)
gx[[tau]]=tx;
gy[[tau]]=ty;
gw[[tau]]=tw
];(*for tau*)
Return[{gxy,gx,gy,gw}
],(*end module/function*)
RuntimeAttributes->{Listable},
Parallelization->False,
CompilationTarget->"C",
RuntimeOptions->"Speed"
];(*end compile*)
<<CompiledFunctionTools` ;
(*CompilePrint[trueCorr]*)
(*yIn=N[{3,5,7,3,8,4,6,7}];Print["yIn=",yIn];
xIn=N[{1,5,3,2,4,6,7,5}];Print["xIn=",xIn];
Print["trueCorr=",trueCorr[xIn,yIn,xIn,yIn]]//AbsoluteTimin
g
g0=trueCorr[r,r,wr,wr]/trueCorr[ones,ones,wr,wr]//AbsoluteT
iming;
Print[ListPlot[g0]];
Print[ListPlot[grr]];*)

(*****
*****
(* define "appendCol"
*)

```

```

(* does NOT expand table if col too long or pad col if too
short *)
(*****
*****
appendCol[a_,b_]:=(*Module[{aa},If[Length[Dimensions[x]]<1,
aa={a};Transpose[Append[aa,b]],
Transpose[Append[Transpose[a],b]]];*)
If[Length[a]!=Length[b],Print["Unequal lengths. Unable to
append"];a,
If[Length[Dimensions[a]]==1,
Join[Transpose[{a}], Transpose[{b}],2],
Join[a, Transpose[{b}],2]]];

(*****
*****
(* define "addParam"
*)
(* if new var runs below existing cols, table is padded
line of blanks*)
(*****
*****
ClearAll[addParam];
addParam[a_,nameCol_,valCol_,name_,val_] :=
Module[{nParams,ap},
nParams=0;
Do[If[a[[i,nameCol]]!=
"",nParams=nParams+1],{i,2,Length[a]}];

If[Length[a]==nParams+1,ap=Append[a,Table["",{i,Length[a][[1
]]}]]];,ap=a];
ap=ReplacePart[ap,{nParams+2,nameCol}-> name];
ap=ReplacePart[ap,{nParams+2,valCol}-> val];
Return[ap]
];

(*****
*****
(* define "mlpNeg" =-Log prob of getting negative v or h
*)
(*****
*****
ClearAll[mlpNeg];
mlpNeg[b_,sb_,v_]:=Log[2]-Log[Erfc[(b-v)/( $\sqrt{2}$  sb)]];

(*****
*****
(* define extrapolate xc or ac curve to time zero

```

```

*)
(*****
*****)
ClearAll[extrap];extrap[x_]:= {x[[1]],x[[1]]-x[[2]],x[[1]]-
2x[[2]]+x[[3]]};

(*****
*****)
(* Fit to single exponential decay
*)
(* mode 0 □ 2p fit all pts & print
*)
(* mode 1 □ 3p fit all pts & print
*)
(* mode 2 □ 2p fit 3 pts & NO print
*)
(*****
*****)
ClearAll[fitExpDecay];
fitExpDecay[iMode_,fitData_,
fitDataName_,wtData_,nEff_,kFirst_,kLast_,fitPlotName_]:=
Module[

{gTable,wTable,iPass,g0,g∞,gTau,t,eq,vars(*,nlm,StepMonitor
,Weights,

VarianceEstimatorFunction,MaxIterations,paramTable,params,e
rrs,estdVar,estdSD,

gSD,gZero,gInf,gTd,gInfErr,gZeroErr,gTdErr,fluctCalc,xyCalc
,range,fitPlot}*)},
saveIPrint[];
iPrint=0;
If[kLast>nEff,gPrint["kLast=",kLast," > nEff=",nEff,".
Results unpredictable"]];
gTable=Table[{i-1,fitData[[i]]},{i,nEff}];
wTable=Table[wtData[[i]},{i,nEff}];

(*gPrint[nEff];
gPrint[wTable];*)
(*gTable=Table[{i-
1,RandomVariate[NormalDistribution[0,2]]},{i,nEff}];
wTable=Table[1,{i,nEff}];*)
Do[If[k<kFirst||k>kLast,wTable[[k]]=0},{k,nEff}];
wTableNorm=(kLast-kFirst+1)*wTable/Total[wTable];
(*Mathematica wts must be norm'd*)
iPass=0;

```

```

Monitor[
  nlm=NonlinearModelFit[
    gTable, (*data as x,y pairs*)
    If[iMode==1(*s*),
      eq=g $\infty$ +(g0-g $\infty$ )*Exp[-t/gTau];
      vars={g0,g $\infty$ ,gTau},
      If[(iMode==0)(* fit all grr*)||(iMode==2)(*fit 3
grr*),
        eq=g0*Exp[-t/gTau];
        vars={g0,gTau},
        gPrint["unrecognized iMode"]; Abort[
]];
      eq, (* eq to use *)
      vars, (* adjustable params*)
      t, (* var in eq*)
      Weights->wTableNorm,
      StepMonitor :>{iPass=iPass+1},
      VarianceEstimatorFunction->(Total[#12 #2]/Total[#2]
n/(n-2)&),
      MaxIterations->1000; (*Infinity*)
      WorkingPrecision->Automatic (*MachinePrecision*)
    ],
    Pause[0.0];
    "iPass="<>ToString[gF[iPass]]<>"
chiSq="(*<>ToString[gF[redChiSq]]*)<>"
tDecay="<>ToString[gF[gTau]]<>"
gInf="<>ToString[gF[g0]]<>ToString[gF[g $\infty$ ]]<>" gAmp="
];

gPrint["\n","fit results"];
gPrint["paramTable=",paramTable=nlm["ParameterTable"]];
(*val,SE,p*)
gPrint["params=",params=nlm["BestFitParameters"]];
gPrint["errs=",errs=nlm["ParameterErrors"]];

gPrint["estdSD=",estdSD=\[Sqrt]nlm["EstimatedVariance"]];
gPrint["nlm[0]=",nlm[0]]; (*"FittedModel"*)

gSD=estdSD;
gZero=params[[1,2]];
gZeroErr=errs[[1]](*errs[[2]]*);
If[(iMode==0)|| (iMode==2), (*2p fits*)
  gInf=0; (*params[[2,2]]*);
  gInfErr=0;
  gTd=params[[2,2]];
  gTdErr=errs[[2]],
  If[iMode==1,(*s*)gInf=params[[2,2]]; (*3p fit*)

```

```

    gInfErr=errs [[2]];
    gTd=params[[3,2]];
    gTdErr=errs[[3]],
    Print["unrecognized iMode.  Exiting..."];Abort[
    ];
dP[gZero];
dP[gZeroErr];
If[(iMode==0)|| (iMode==1),
    fluctCalc=Table[nlm[i-1],{i,1,nEff}];
    xyCalc=Table[{i-1,nlm[i-
1]},{i,kLast}];range={1.1Min[xyCalc[[All,2]],gTable[[All,2]
]],1.1*Max[xyCalc[[All,2]],gTable[[All,2]]]};

fitPlot=ListPlot[{gTable[[kFirst;;kLast,2]],xyCalc[[kFirst;
;kLast,2]]},PlotLabel-> fitPlotName]
];(*if*)
restIPrint[];
];
(*setIPrint[0];fitExpDecay[fitData,
fitDataName,wtData,nEff,kFirst,kLast,fitPlotName];restIPrin
t[];*)

(*****
*****
(*getFileList[]:  get names of raw data files in current
directory *)
(*****
*****
ClearAll[getFileList];
getFileList[]:=Module[{f,l,c1,c2,c3,c4,c,i},
    f=FileNames[All];l=Length[f];
    c1=StringContainsQ[f, "LRB"];c2=StringContainsQ[f,
"csv"];
    c3=Thread[!StringContainsQ[f,
"results"]];c4=Thread[!StringContainsQ[f, "plots"]];
    c=Table[c1[[i]]&& c2[[i]]&& c3[[i]]&& c4[[i]],{i,l}];
    fs={};For [i=1,i<=l,i++,If[c[[i]],AppendTo[fs,f[[i]]]];
    Return[fs]
];
(*getFileList[]*)

(*****
*****
(* wtd linear fit
*)
(*****
*****

```

```

ClearAll[probs,xList,data,wtdLin];
wtdLinFit[s_,d_,nP_,iFirstOrder_,wTable_]:=Module[{x,xList,
data,ev, β,se,pr,probs,lm},
  (* nP=num params; iFirstOrder=*)
  n=Length[s];
  data=Transpose@{s,d};
  (*Print[ListPlot[data]];*)
  xList=Table[xi,{i,iFirstOrder,nP+iFirstOrder-1}];
  (*Print[xList];*)
  lm=LinearModelFit [data,xList,x,
    IncludeConstantBasis->False,
    Weights->wTable,
    VarianceEstimatorFunction->((Total[#12
#2]/Total[#2])* (n/(n-2)) &)];
  β=lm["BestFitParameters"];
  ev=lm["EstimatedVariance"];
  se=lm["ParameterErrors"]; (* std err*)
  pr=lm["PredictedResponse"]; (* yCalc*)
  probs= (1/2) (β/se)2;
  (*dP[probs];
  dP[Total[probs]];*)
  Return[{ev,β,se, (1/2) (β/se)2}]
  (* [[1]]= fit variance, [[2]]=params, [[3]]=std errs,
  [[4]]=mlp param probs *)
  ];

  (*****
  **)
  (* functiot to calc variance of variance of list
  *)
  (*****
  **)
ClearAll[varOfVar];
varOfVar[x_]:=Module[{n,
μ,ones,dxSq,μ2,μ4,μ2Sq,varVar,vov},
  n=Length[x];
  μ=Total[x]/n;
  ones=Table[1,{n}];
  dxSq=(x-ones*μ)2;
  μ2=Total[dxSq]/n;
  μ4=Total[dxSq2]/n;
  μ2Sq=μ2;
  vov=(μ4-μ2Sq)/n+μ2Sq*(2/(n(n-1)))
  ];

  (* (*****
  *****)

```

```

(* test var of difference in variances
*)
(*****
**)
ClearAll[x,y,vx,vy,diff,vvx,vvy,vvsum,vvsumsd,ratio];
n=1000;
m=1000;
iPrint=0;
x=RandomVariate[NormalDistribution[10,1],n];
y=RandomVariate[NormalDistribution[10,1],n];
xt=Table[x=RandomVariate[NormalDistribution[10,1],n],
m];(*dP[xt];*)
yt=Table[y=RandomVariate[NormalDistribution[10,1],n],
m];(*dP[yt];*)
s2x=Variance[xt[]];dP[s2x];
s2y=Variance[yt[]];dP[s2y];
vovx=Table[varOfVar[xt[[i]]],{i,m}];dP[vovx];
vovy=Table[varOfVar[yt[[i]]],{i,m}];dP[vovy];
diff=s2x- s2y; dP[diff];
vvsum=vovx+vovy;dP[vvsum];
vvsumsd=Sqrt[vvsum];dP[vvsumsd];
ratio=(diff/vvsumsd);dP[ratio];
mlp=ratio^2/2;dP[mlp];
iPrint=1;
Length[ratio]
Variance[ratio]
Histogram[ratio]*)

restIPrint[]; (* overall for 'definitions'*)

(*****
**)
(* end of definitions
*)
(*****
**)

ClearAll[depend];
depend[vEff_List,hEff_List,bvEff_List,bhEff_List,iWt_Integer,
iGeom_Integer,kFirst_Integer,kLast_Integer,nBkgsDummy_Integer,
gDummy_Real,bDummy_Real,cDummy_Real,fDummy_Real,iFirstDot_Integer,
iLastDot_Integer]:=Module[{{(*nMax,n,m,nEff,nCorr,kMax,kMin,nk,nBkgs*)}},

(*****
*****
*)
(* initialization code

```

```

*) (*****
*****)
  saveIPrint[];
  (*dP[isNewMin," ", iPass]; *)
  If [Length[vEff]!= Length[hEff], Print["vEff and hEff
unequal length. Exiting..."];Abort[] ];
  nEff=Length[vEff];
  zeros=Table[0,{nEff}];
  ones=Table[1,{nEff}];
  kList=Table[nEff-k+1,{k,1,nEff}];
  sqrtKList= $\sqrt{kList}$  ;
  time=Table[i, {i,nEff}];
  kMax=If[kLast==0, nEff, kLast];
  If [kLast>nEff,Print["kMax exceeds nEff.
Exiting..."];Abort[]];
  kMin=If[kFirst==0, 2,kFirst];
  nk=kMax-kMin+1;
  (*nBkgs=If [nBkgsDummy<0,4,nBkgsDummy];*)

(*****
*****)
  (* set constants
*)

(*****
*****)
  g=gDummy; (* old g-fact: (1-g)v, (1+g)h *)
  b=bDummy; (* sample bkg *)
  c=cDummy; (* camera bkg *)
  f=fDummy; (* ???? *)
  (*Pause[1*^9];*)

(*****
*****)
  (* CALCULATION STARTS HERE
*)

(*****
*****)

(*****
*****)
  (* get inten funct & baseline vars
*)

```

```

(*****
****)
  (* nBkgs = number of baselines averaged *)
  bvAvg=Mean[bvEff]; (* avg of nBkgs traces *)
  bhAvg=Mean[bhEff];

  If[ f<0,f=cBkg];(*if f<0 (e.g. -1), f initialized to
cBkg *)
  If [isBkgs!=0, (*yes, bkgs. calc inten funct from bkgs
*)
    intV=(bvEff-f)/(bvAvg-f);(*inten should=iv,ih*)
    intH=(bhEff-f)/(bhAvg-f),(*inten should=iv,ih*)
    (*else let inten factors=1 *)
    intV=Table[1,{nEff}];
    intH=intV
  ];
  (*dP[intH];
Pause[1*^9];*)

  bvc=(1-g)bvEff/intV; (* correcting for src fluctu *
*)
  bhc=(1+g)bhEff/intH; (* no corr if no bKgs *)

  If [isBkgs!=0,(*yes, bkgs. calc bkg variances
UNCORRECTED *)
    e2bv=Variance[bvEff]; (* actual variance corr'd bkgs *)
    e2bh=Variance[bhEff],
    (*else*)
    e2bv=1; (* need some placeholder num here *)
    e2bh=1
  ];

  If [isBkgs!=0,(*yes, bkgs. calc CORRECTED bkg variances
*)
    e2bvc=(1-g)^2 Variance[bvEff]; (* actual variance corr'd
bkgs *)
    e2bhc=(1+g)^2 Variance[bhEff],
    (*else*)
    e2bvc=1; (* need some placeholder num here *)
    e2bhc=1
  ];

  If [iWt==2, (* rs weights*)
    (*Gaussian*)vVar=e2bvc;hVar=e2bhc,
    If[iWt==1,
      (*Poisson*)vVar=(1-g)^2 (vEff+2 bvEff); (* 2 baselines /
vc*)hVar=(1+g)^2 (hEff+2 bhEff),

```

```

Print["bad iWt"]]
];
vVarSq=vVar2;
hVarSq=hVar2;

(*intV=(bvEff-f)/(bvAvg-f);(*inten should=iv,ih*)
intH=(bhEff-f)/(bhAvg-f);*)

vs=(vEff-bvEff)/intV; (* v sub't'd *)
hs=(hEff-bhEff)/intH;
vM=Mean[vs]; (* v(sub't'd)Mean *)
hM=Mean[hs];
vSD=StandardDeviation[vs];
hSD=StandardDeviation[hs];

g= (vSD-hSD)/(vSD+hSD);
c=(hM(1+g)-(1-g) vM)/(2g);
(*Print[iPass," ",vM," ",hM," ",vSD," ",hSD," ",g,"
",c," ",b];*)
vc=Collect[Simplify[(1-g)(vs-c)-b/2],vs];
hc=Collect[Simplify[(1+g)(hs-c)+b/2],hs];

s=vc+iGeom*hc;sSq=s2;s4th=sSq2;
d=vc-hc;dSq=d2;d4th=dSq2;
r=Table[If[s[[i]]!=0,d[[i]]/s[[i]],0], {i,nEff}];rSq=r2;

(*****
****)
(* get variances and wts
*)

(*****
****)
If[(isNewMin!=0)|| (iPass==0 ),(*dP[iPass];*)
(*Print[iPass];*)
sVar=ones*(2(vVar+iGeom2 hVar)); (* two bkgs each
meas't*)
dVar=ones*(2(vVar+hVar))
];

(*get INITIAL values for sAvg, rAvg to calc init rWts*)
If[iPass== 0,
ws=ones/sVar;
sWS=s*ws;
sAvg=Total[sWS]/Total[ws];
rAvg=zeros (* start with rAvg = 0*)

```

```

];

wsRaw=ones/sVVar;
ws=nEff*wsRaw/Total[wsRaw]; (*normalize s-wts*)
sws=s*ws;
sSqws=sSq*ws;
sAvg=Total[sws]/Total[ws];

wdRaw=ones/dVVar;
wd=nEff*wdRaw/Total[wdRaw]; (*normalize d-wts*)
dwd=d*wd;
dSqwd=dSq*wd;
dAvg=Total[dwd]/Total[wd];

(* only have 2nd order sums of v, h so ignore 4th-order
tems *)
z=iGeom;rVar=(1/s4th)*(vVar (-1+r)^2 sSq+hVar sSq (1+r
z)^2 +3 vVarSq (-1+r)^2+3 z^2 hVarSq (1+r z)^2 +3 vVar hVar
((1-4 z+z^2)+6 r z (1-z) +6 z^2 rSq));wrRaw=1/rVar;
wr=nEff*wrRaw/Total[wrRaw];(*normalize r-wts*)
rwr=r*wr;
rSqwr=rSq*wr;
rAvg=Total[rwr]/Total[wr];

(*Print["end calc basic terms"];*)

(*****
***)
(* calc correlations (now AVERAGES )
*)

(*****
***)

(*kList=Table[nEff-k+1,{k,1,nEff}];*)
(*oneList=Table[1,{k,1,nEff}];*)
ClearAll[];
sws=s*ws;
rwr=r*wr;
sAvg=Total[sws]/Total[ws];
rAvg=Total[rwr]/Total[wr];
(*g11=ListCorrelate[ones,ones,1,0];*)

gss=ListCorrelate[sws,sws,1,0];
gs1=ListCorrelate[sws,ones,1,0];
g1s=ListCorrelate[ones,sws,1,0];

```

```

gsws=ListCorrelate[ws,ws,1,0];
gws1=ListCorrelate[ws,ones,1,0];
glws=ListCorrelate[ones,ws,1,0];
gssFluct12=gss/gsws-gs1/gws1 gls/glws ;

gsr=ListCorrelate[sws,rwr,1,0];
(*gs1=ListCorrelate[sws,ones,1,0];*) (* already done *)
glr=ListCorrelate[ones,rwr,1,0];
gswr=ListCorrelate[ws,wr,1,0];
(*gws1=ListCorrelate[ws,ones,1,0];*) (* already done *)
glwr=ListCorrelate[ones,wr,1,0];
gsrFluct12=gsr/gswr-gs1/gws1 glr/glwr ;

grs=ListCorrelate[rwr,sws,1,0];
gr1=ListCorrelate[rwr,ones,1,0];
(*gl1=ListCorrelate[ones,sws,1,0];*) (* already done *)
gwrws=ListCorrelate[wr,ws,1,0];
gwrl=ListCorrelate[wr,ones,1,0];
(*glws=ListCorrelate[ones,ws,1,0];*) (* already done *)
grsFluct12=grs/gwrws-gr1/gwrl gls/glws ;

grr=ListCorrelate[rwr,rwr,1,0];
(*gr1=ListCorrelate[rwr,ones,1,0];*) (* already done *)
(*glr=ListCorrelate[ones,rwr,1,0];*) (* already done *)
gwrwr=ListCorrelate[wr,wr,1,0];
(*gwrl=ListCorrelate[wr,ones,1,0];*) (* already done *)
(*glwr=ListCorrelate[ones,wr,1,0];*) (* already done *)
grrFluct12=grr/gwrwr-gr1/gwrl glr/glwr;

(* (***** enforce that vVar = hVar
*****)
x=StandardDeviation [vs];
y=StandardDeviation[hs];
g=(x-y)/(x+y);*)

(* (***** prob the neg pts are really zero
*****)
(* assume true val of -'ve pt actually zero *)
mlpNegVSum=Total[0.5*(1-Sign[vc])*vc2/(2*e2bvc)];
mlpNegHSum=Total[0.5*(1-Sign[hc])*hc2/(2*e2bhc)];
mlpNeg=mlpNegVSum+ mlpNegHSum; (* vc,hc must both be
non-neg *)
(*Print[vc[[10]]," ",mlpNegVSum," ",mlpNegHSum,"
",mlpNeg, " iWt=",iWt];*)

(* (***** prob that s,r uncorrelated

```

```

*****
(* all "mlp" terms preceded by (1/2)Log[2 π]-
(1/2)Log[nEff] *)
(* distCorr=distCorrFunct[r,s];*)

(*****
****)
RSDistCorr=distCorrFunct[r,s];
RSPmCorr= wtdCorrCoeff[r,s,ones];
RSWpmCorr= wtdCorrCoeff[r,s,wr*ws];
mlpRSDistCorr=(1/2)RSDistCorr/(1(*-rho^2*)) nEff;
mlpRSPmCorr=(1/2)RSPmCorr^2 / (1(*-rho^2*)) nEff;
mlpRSWpmCorr=(1/2)RSWpmCorr^2/( 1(*-rho^2*)) nEff;

(* (***** prob coeffs a0=0,a1=0 ***** )
wTable=ws*wr;
mlpRAvg=wtdLinFit[s,r,1,0,wTable][[4]][[1]];
mlpRSlope=wtdLinFit[s,r,1,1,wTable][[4]][[1]];
(*mlpRCurve=wtdLinFit[s,r,1,2,wTable][[4]][[1]];*)
(*mlpR4th=wtdLinFit[s,r,1,3,wTable][[4]][[1]];*)*)
wTable=ws*wr;
mlpRAvg=0;
mlpRSlope=wtdLinFit[s,r,1,1,wTable][[4]][[1]];
mlpRCurve=0;

(* (***** enfoprob that v- & h-variances are same
*****)
varDiff=Variance[vc]-Variance[hc];
varDiffErr=Sqrt[(varOfVar[vc]+varOfVar[hc])/1(*only 1
pt*)];
mlpVarDiff=(1/2)(varDiff/varDiffErr)^2; (* d=(vc-bvc)-
(hc-bhc) OK!!!! *)
(*Print["dAvg=", dAvg," dAvgErr=",dAvgErr,"
mlpDAvg1P=",mlpDAvg1P];*)*)
mlpVarDiff=0;

(* (***** prob that true min is zero ***** )
nMin=16;
vcs=Sort[vc];
hcs=Sort[hc];
mlpZeroErr=Sum[vcs[[i]]^2/(2*varV)+
hcs[[i]]^2/(2*varH),{i,1,nMin}];*)
mlpZeroErr=0;

(***** overall probability ***** )
(* all selected measures *)

```

```

    mlpTot=0;
    mlpTot=mlpTot+0mlpNeg;          (*prob neg pts are really
zero*)
    mlpTot=mlpTot+0mlpRSDistCorr;   (*prob that r,s
uncorrelated*)
    mlpTot=mlpTot+0mlpRSPmCorr;     (*prob that r,s
uncorrelated; wtd*)
    mlpTot=mlpTot+1mlpRSWpmCorr;    (*prob that r,s
uncorrelated; wtd*)

    mlpTot=mlpTot+0 mlpRAvg;        (*prob that intercept is
zero*)
    mlpTot=mlpTot+0mlpRSlope;       (*prob that slope is zero*)
    mlpTot=mlpTot+0mlpRCurve;       (*prob that slope is zero*)
    (*mlpTot=mlpTot+1mlpRCurve;*)   (*prob that slope is
zero*)

    mlpRCurve=mlpZeroErr;
    mlpTot=mlpTot+0 mlpVarDiff;
    mlpTot=mlpTot+0 mlpZeroErr;

    mlpTot=mlpTot(*+0 p3*);        (*max of 3 adj constants*)

    Return[mlpTot]
]; (*module*)
(*inFileName="Ige-1-LRB.csv";
data=Import[inFileName];
bvEff=data[[2;;All,2]];
bhEff=data[[2;;All,3]];
vEff=data[[2;;All,8]]; hEff=data[[2;;All,9]];
dep=depend[vEff,hEff,bvEff,bhEff,2,1, 0,0,4,+0.4,0.,0.,-
1.,iFirstDot,iLastDot];
Print["nEff=",nEff, " dep=", dep];
isNewMin=1;
depend[];*)

restIPrint[];

(*****
***)
(* end of "depend" code
*)
(*****
***)

(*****
*****)

```

```

(* maximize prob wrt g,b,c
*)
(*****
*****
ClearAll[optimize ];
optimize[]:=Module[>(*iWt=2,
iFirst=1,iLast=1000,kFirst=0,nBkgs=4,kLast=1000,iThreshMode
=0,thresh=0,iOrderSave=4,iGeom=1,g0=.40,b0=0.,c0=0.,f0=-
1.*)],

    saveIPrint[];
    (*progressFileName=base<>"-"<>"progress"<>"-
"<>DateString[]<>".csv";
    If [FileExistsQ[progressFileName],
    Close[progressFileName];
    DeleteFile[progressFileName]];
    prog=OpenWrite[progressFileName]; *)

pInit=depend[vEff,hEff,bvEff,bhEff,iWt,iGeom,0,0,nBkgs,g0,b
0,c0,f0,iFirstDot,iLastDot];
    (*Print["pInit=",pInit," g0=",g0," b0=", b0,"
c0=",c0," f0=", f0," iWt=",iWt];*)

    pMin=pInit;
    iPass=0;
    qq={0,{{0,0},{0,0},{0,0},{0,0}}};
    Off[StringJoin::string];
    c=c0;
    f=f0;
    isNewMin=1;
    qq=Monitor[

FindMinimum[{depend[vEff,hEff,bvEff,bhEff,iWt,iGeom,0,0,nBk
gs,g,b,c,f,iFirstDot,iLastDot],(*b≥ 0&&c≥ 0&&*) -1<g<1(*&&
-1<f<1*)},{(*{g,g0}*)(*,*){b,b0}{*,{c,c0}*)(*,{f,f0}*)},
    EvaluationMonitor:>
    (
        p=mlpTot;
        iPass=iPass+1;
        (*WriteString[prog,
gF[iPass],",",gF[p],",",gF[mlpNeg],",",gF[mlpBkgs],",",gF[m
lpRAvg],",",gF[mlpDAvg],",",gF[mlpRhoSR],",",gF[mlpRhoRS],",
",gF[mlpRhoSSR],",",gF[mlpRhoSRR],",",gF[g],",",gF[c],",",g
F[b],"\n"];*)
        If[p<pMin ,
            gMin=g;bMin=b;cMin=c;pMin=p;fMin=f;

```

```

        isNewMin=1(*;dP[iPass]*),
        (*else*) isNewMin=0;];
    If[p<pGoal|| iPass>= maxPass, Goto[continue]];
    If
[CurrentValue["ControlKey"]==True,Goto[continue]]
    ),(* end EvalMon*)
    PrecisionGoal->Automatic (* tried 0 *),
    AccuracyGoal ->Automatic (* tried 3 *),
    MaxIterations->1000(*Infinity*),
    WorkingPrecision->Automatic (*"MachinePrecision"*),
    Method->Automatic
    ], (* end FindMin*)
    "iPass="<>ToString[gF[iPass]]
    <>" iDot=" <>ToString[gF[iDot]]
    <>" of " <>ToString[gF[nDots]]

    <>"\n RAvg="<> ToString[gF[mlpRAvg]]
    <>" RSlope="<> ToString[gF[mlpRSlope]]
    <>" RCurve="<> ToString[gF[mlpRCurve]]
    <>" varDiff="<> ToString[gF[mlpVarDiff]]
    <>" zeroErr="<> ToString[gF[mlpZeroErr]]

    <> "\n mlpNeg=" <> ToString[gF[mlpNeg]]
    <> " RSDistCorr=" <> ToString[gF[mlpRSDistCorr]]
    <> " RSPmCorr=" <> ToString[gF[mlpRSPmCorr]]
    <> " RSWPmCorr=" <> ToString[gF[mlpRSWpmCorr]]

    <>"\n p="<>ToString[gF[p]]
    <>" g="<>ToString[gF[g]]
    <> " b="<>ToString[gF[b]]
    <> " c="<>ToString[gF[c]]
    <> " f="<>ToString[gF[f]]

]; (*end Monitor*)

Label[continue ];
(*Print["\n iPass=",iPass," pMin=",pMin," gMin=",gMin,"
bMin=",bMin," cMin=", cMin, " fMin=",fMin];*)
(*Print["pause"];
Pause[1*^6];*)

If[qq[[1]]==0&& p>pGoal,
Print["Optimization NOT completed! " ],
(*otherwise*)
Print["Optimization successful! - p = ",p]
];

```

```

p=depend[vEff,hEff,bvEff,bhEff,iWt,iGeom,kFirst,kLast,nBkgs
,gMin,bMin,cMin,fMin,iFirstDot,iLastDot];
Print["iPass="<>ToString[gF[iPass]]
  <>" iDot=",ToString[gF[iDot]]
  <>" of " <>ToString[gF[nDots]]

  <>"\n RAvg="<> ToString[gF[mlpRAvg]]
  <>" RSlope="<> ToString[gF[mlpRSlope]]
  <>" RCurve="<> ToString[gF[mlpRCurve]]
  <>" varDiff="<> ToString[gF[mlpVarDiff]]
  <>" zeroErr="<> ToString[gF[mlpZeroErr]]

  <>"\n mlpNeg=" <> ToString[gF[mlpNeg]]
  <>" mlpRSDistCorr=" <> ToString[gF[mlpRSDistCorr]]
  <>" mlpRSPmCorr=" <> ToString[gF[mlpRSPmCorr]]
  <>" mlpRSWpmCorr=" <> ToString[gF[mlpRSWpmCorr]]

  <>"\n p="<>ToString[gF[p]]
  <>" g="<>ToString[gF[g]]
  <>" b="<>ToString[gF[b]]
  <>" c="<>ToString[gF[c]]
  <>" f="<>ToString[gF[f]]
];
Print[{vchcPlot=ListPlot[{vc,hc},PlotLabel->"vc hc"],
  vchcDiffPlot=ListPlot[{vc-hc},PlotLabel->"vc-hc"],
  sPlot=ListPlot[{s},PlotLabel->"s"],
  rPlot=ListPlot[{r},PlotLabel->"r"]} ];
restIPrint[];

b=Table[i,{i,0,150,2}];
(*Print[{Histogram[vc,{b}],Histogram[vc,{b}]}]*)
];
(*optimize[];*)

(*****
*****)
(*****
*****)
(*****
*****)
(*****
*****)
(*****
*****)
(*****
*****)
(*****
*****)
(*****
*****)

```



```

bNew=bMin;
cNew=cMin;
fNew=fMin;

(*depend[vEff_List,hEff_List,bvEff_List,bhEff_List,iWt_Integer,iGeom_Integer,kFirst_Integer,kLast_Integer,nBkgsDummy_Integer,gDummy_Real,bDummy_Real,cDummy_Real,fDummy_Real,iFirstDot_Integer,iLastDot_Integer*)
(Print["pOpt=",depend[vEff,hEff,bvEff,bhEff,iWt,iGeom,kFirst,kLast,nBkgs,gNew,bNew,cNew,fNew,iFirst,iLastDot],
" gNew=",gNew," bNew=", bNew," cNew=",cNew," fNew=",fNew]*);
SetOptions[{Plot,ListPlot},ImageSize->{144,108}];
ListPlot[{vc,hc},PlotRange->Automatic(*{0,100}*),PlotLabel->"vc,hc"];

gsrPlots={gsrFullPlot=ListPlot[{gsrFluct12[[2;;]]},PlotLabel->"gsrFluctFull"],
gsr500Plot=ListPlot[{gsrFluct12[[2;;500]]},PlotLabel->"gsrFluct500"],
gsr200Plot=ListPlot[{gsrFluct12[[2;;200]]},PlotLabel->"gsrFluct200"],
gsr100Plot=ListPlot[{gsrFluct12[[2;;100]]},PlotLabel->"gsrFluct100"]};
gPrint[gsrPlots];

grsPlots={grsFullPlot=ListPlot[{grsFluct12[[2;;]]},PlotLabel->"grsFluctFull"],
grs500Plot=ListPlot[{grsFluct12[[2;;500]]},PlotLabel->"grsFluct500"],
grs200Plot=ListPlot[{grsFluct12[[2;;200]]},PlotLabel->"grsFluct200"],
grs100Plot=ListPlot[{grsFluct12[[2;;100]]},PlotLabel->"grsFluct100"]};
gPrint[grsPlots];

gssPlots={gssFullPlot=ListPlot[{gssFluct12[[2;;]]},PlotLabel->"gssFluctFull"],
gss500Plot=ListPlot[{gssFluct12[[2;;500]]},PlotLabel->"gssFluct500"],
gss200Plot=ListPlot[{gssFluct12[[2;;200]]},PlotLabel->"gssFluct200"],
gss100Plot=ListPlot[{gssFluct12[[2;;100]]},PlotLabel->"gssFluct100"]};

```

```

gPrint[gssPlots];

grrPlots={grrFullPlot=ListPlot[{grrFluct12[[2;;]]}],PlotLabel->"grrFluctFull"},
grr500Plot=ListPlot[{grrFluct12[[2;;500]]}],PlotLabel->"grrFluct500"},
grr200Plot=ListPlot[{grrFluct12[[2;;200]]}],PlotLabel->"grrFluct200"},
grr100Plot=ListPlot[{grrFluct12[[2;;100]]}],PlotLabel->"grrFluct100"};
Print[grrPlots];
restIPrint[];
);
(*showOptResults[]*)

(*****
*****
(*ANIS: fit anis decays of various ranges
*)
(*****
*****
ClearAll[fitAnisDecay];
fitAnisDecay[]:= (
saveIPrint[];
iPrint=0;
grFitData=grrFluct12;
grFitDataName="grrFluct";
wrr=gwrwr;
ClearAll[grFitPlots,grFitPlotFull,grFitPlot500,
grFitPlot200, grFitPlot100];

grFitFull=fitExpDecay[0,grFitData,
grFitDataName,wrr,nEff,2,Length[grFitData]-1,"grFitFull"];
grkFirstFull=2;grkLastFull=Length[grFitData];

grSDFull=gSD;grInfFull=gInf;grZeroFull=gZero;grTdFull=gTd;

grInfErrFull=gInfErr;grZeroErrFull=gZeroErr;grTdErrFull=gTdErr;
grFluctCalcFull=fluctCalc;grFitPlotFull=fitPlot;

grFit500=fitExpDecay[0,grFitData,
grFitDataName,wrr,nEff,2,500,"grFit500"];
grkFirst500=2;grkLast500=500;
grSD500=gSD;grInf500=gInf;grZero500=gZero;grTd500=gTd;

```

```

grInfErr500=gInfErr;grZeroErr500=gZeroErr;grTdErr500=gTdErr
;
  grFluctCalc500=fluctCalc;grFitPlot500=fitPlot;

  grFit200=fitExpDecay[0,grFitData,
grFitDataName,wrr,nEff,2,200,"grFit200"];
  grkFirst200=2;grkLast200=200;
  grSD200=gSD;grInf200=gInf;grZero200=gZero;grTd200=gTd;

grInfErr200=gInfErr;grZeroErr200=gZeroErr;grTdErr200=gTdErr
;
  grFluctCalc200=fluctCalc;grFitPlot200=fitPlot;

  grFit100=fitExpDecay[0,grFitData,
grFitDataName,wrr,nEff,2,100,"grFit100"];
  grkFirst100=2;grkLast100=100;
  grSD100=gSD;grInf100=gInf;grZero100=gZero;grTd100=gTd;

grInfErr100=gInfErr;grZeroErr100=gZeroErr;grTdErr100=gTdErr
;
  grFluctCalc100=fluctCalc;grFitPlot100=fitPlot;

  Print[grFitPlots={grFitPlotFull,grFitPlot500,
grFitPlot200, grFitPlot100}];
  restIPrint[];
  );
(*fitAnisDecay[]*)

(*****
*****
(*ANIS:  fit single exponential decay - 3 parameters
*)
(*****
*****)
fitAnisDecay3Pts[:=(
  (*gTable=Table[{i-1,fitData[[i]},{i,nEff}];*)
  saveIPrint[];
  gTable={grrFluct12
[[2]],grrFluct12[[3]],grrFluct12[[4]]};
  wTable={1,1,1};
  fitExpDecay[2(*iMode*),gTable,
Null(*fitDataName*),wTable,3,1,3,Null];
  gZero3Pts=gZero;
  gZeroErr3Pts=gZeroErr;
  gTd3Pts=gTd;
  gTdErr3Pts=gTdErr;
  (*nums[[127]]=Log10[gTd];

```

```

      nums[[128]]=(0.434294/gTd)*gTdErr;*)
      gSD3Pts=gSD);

(*****
*****
(*INTEN:  fit single exponential decay
*)
(*****
*****

ClearAll[fitIntenDecay];
fitIntenDecay[:=(
  saveIPrint[];
  iPrint=0;
  gsFitData=gssFluct12;
  gsFitDataName="gssFluct";
  wss=gswsws;

  (*iMode=r - force fitted flucTs to go to zero *)
  gsFitFull=fitExpDecay[0,gsFitData,
gsFitDataName,wss,nEff,2,nEff,"gsFitFull"];
  gskFirstFull=2;gskLastFull=999;

gsSDFull=gSD;gsInfFull=gInf;gsZeroFull=gZero;gsTdFull=gTd;

gsInfErrFull=gInfErr;gsZeroErrFull=gZeroErr;gsTdErrFull=gTd
Err;
  gsFluctCalcFull=fluctCalc;gsFitPlotFull=fitPlot;

  gsFit500=fitExpDecay[0,gsFitData,
gsFitDataName,wss,nEff,2,500,"gsFit500"];
  gskFirst500=2;gskLast500=500;
  gsSD500=gSD;gsInf500=gInf;gsZero500=gZero;gsTd500=gTd;

gsInfErr500=gInfErr;gsZeroErr500=gZeroErr;gsTdErr500=gTdErr
;
  gsFluctCalc500=fluctCalc;gsFitPlot500=fitPlot;

  gsFit200=fitExpDecay[0,gsFitData,
gsFitDataName,wss,nEff,2,200,"gsFit200"];
  gskFirst200=2;gskLast200=200;
  gsSD200=gSD;gsInf200=gInf;gsZero200=gZero;gsTd200=gTd;

gsInfErr200=gInfErr;gsZeroErr200=gZeroErr;gsTdErr200=gTdErr
;
  gsFluctCalc200=fluctCalc;gsFitPlot200=fitPlot;

```

```

    gsFit100=fitExpDecay[0,gsFitData,
gsFitDataName,wss,nEff,2,100,"gsFit100"];
    gskFirst100=2;gskLast100=100;
    gsSD100=gSD;gsInf100=gInf;gsZero100=gZero;gsTd100=gTd;

gsInfErr100=gInfErr;gsZeroErr100=gZeroErr;gsTdErr100=gTdErr
;
    gsFluctCalc100=fluctCalc;(gsFitPlot100=fitPlot);

    gPrint[gsFitPlots={gsFitPlotFull,gsFitPlot500,
gsFitPlot200, gsFitPlot100}];
    restIPrint[];
    );

(*setIPrint[0];fitIntenDecay[];restIPrint[];*)

(*****
*****)
(* save results as CSV
*)
(*****
*****)

ClearAll[saveResults];
saveResults[:=(
    (* generate timestamp for all output this pass *)
    saveIPrint[0];
    isStamp=True;
    base=StringTake[inFileName,StringLength[inFileName]-
4]<>If[iDot!=-1,"-
d"<>IntegerString[iDot,10,2],""];tdList=DateList[];
    dateFormat={"Year","-","Month","-","Day"};
    timeFormat={"Hour",":","Minute",":","Second"};
    dateString=DateString[tdList,dateFormat];
    timeString=DateString[tdList,timeFormat];
    timeStampFormat={"*-","*")"Year",(*"-","*")"Month",(*"-
",*)"Day", "-","Hour24","Minute","Second"};

timeStampString=If[isStamp,DateString[tdList,timeStampForma
t],""];
    (* dP[timeStampString];*)

    (* set up results file *)
    resultsFileName=base<>"- "<>"results"<>"-
"<>timeStampString<>".csv";

    If [FileExistsQ[resultsFileName],

```

```

q=Close[resultsFileName];
gPrint[q];
DeleteFile[resultsFileName]
];
res=OpenWrite[resultsFileName];
nameCol=1;
valCol=2;

ClearAll[rA]; (* rA= "Results Array" *)
(* fill columns C through V with all pts *)
rA=Prepend[Table["",{i,nEff}], "param"];
rA=appendCol[rA,Prepend[Table["",{i,nEff}], "value"]];
rA=appendCol[rA,Prepend[Table[i,{i,nEff}], "iPt"]];
rA=appendCol[rA,Prepend[vEff, "vEff"]];
rA=appendCol[rA,Prepend[hEff, "hEff"]];
rA=appendCol[rA,Prepend[bvEff, "bvEff"]];
rA=appendCol[rA,Prepend[bhEff, "bhEff"]];
rA=appendCol[rA,Prepend[vc, "vc"]];
rA=appendCol[rA,Prepend[hc, "hc"]];
rA=appendCol[rA,Prepend[s, "s"]];
rA=appendCol[rA,Prepend[d, "d"]];
rA=appendCol[rA,Prepend[r, "r"]];
rA=appendCol[rA,Prepend[wr, "wr"]];
(*rA=appendCol[rA,Prepend[gssFluct, "gssFluct"]];*)
rA=appendCol[rA,Prepend[gssFluct12, "gssFluct12"]];
(*rA=appendCol[rA,Prepend[gsrFluct, "gsrFluct"]];*)
rA=appendCol[rA,Prepend[gsrFluct12, "gsrFluct12"]];
(*rA=appendCol[rA,Prepend[grsFluct, "grsFluct"]];*)
rA=appendCol[rA,Prepend[grsFluct12, "grsFluct12"]];
(*rA=appendCol[rA,Prepend[grrFluct, "grrFluct"]];*)
rA=appendCol[rA,Prepend[grrFluct12, "grrFluct12"]];
rA=appendCol[rA,Prepend[wrr, "wrr"]];
rA=appendCol[rA,Prepend[grFluctCalcFull,
"grrFluctCalcFull"]];
rA=appendCol[rA,Prepend[grFluctCalc500,
"grrFluctCalc500"]];
rA=appendCol[rA,Prepend[grFluctCalc200,
"grrFluctCalc200"]];
rA=appendCol[rA,Prepend[grFluctCalc100,
"grrFluctCalc100"]];

(* fill cols A and B with names and values of params
*) rA=addParam[rA,nameCol,valCol,"nb",NotebookFileName[]]; rA
=addParam[rA,nameCol,valCol,"inFileName",inFileName];
rA=addParam[rA,nameCol,valCol,"iDot",iDot];
rA=addParam[rA,nameCol,valCol,"nEff",nEff];
rA=addParam[rA,nameCol,valCol,"cBkg",cBkg];

```

```

    rA=addParam[rA,nameCol,valCol,"thMode",0]; (* no longer
used*)
    rA=addParam[rA,nameCol,valCol,"thLev",0]; (* no longer
used*)
    rA=addParam[rA,nameCol,valCol,"nCorr",0]; (* no longer
used*)
    rA=addParam[rA,nameCol,valCol,"n2v",0]; (* no longer
used*)
    rA=addParam[rA,nameCol,valCol,"n2h",0]; (* no longer
used*)
    rA=addParam[rA,nameCol,valCol,"n2vc",0]; (* no longer
used*)
    rA=addParam[rA,nameCol,valCol,"n2sc",0]; (* no longer
used*)

    rA=addParam[rA,nameCol,valCol," ", " "];
    rA=addParam[rA,nameCol,valCol,"kMin",kMin];
    rA=addParam[rA,nameCol,valCol,"kMax",kMax];

rA=addParam[rA,nameCol,valCol,"mlpIntercept",mlpRAvg(*mlpIn
tercept*)];

rA=addParam[rA,nameCol,valCol,"mlpSlope",mlpRSlope(*mlpSlop
e*)];

rA=addParam[rA,nameCol,valCol,"mlpLFeedthru",mlpVarDiff];
    rA=addParam[rA,nameCol,valCol,"mlpTot",mlpTot];
    rA=addParam[rA,nameCol,valCol,"unused1",0];
    rA=addParam[rA,nameCol,valCol,"unused2",0];
    rA=addParam[rA,nameCol,valCol,"iPass",iPass];
    rA=addParam[rA,nameCol,valCol,"pMin",pMin];
    rA=addParam[rA,nameCol,valCol,"gMin",gMin];
    rA=addParam[rA,nameCol,valCol,"bMin",bMin];
    rA=addParam[rA,nameCol,valCol,"cMin",cMin];
    rA=addParam[rA,nameCol,valCol,"fMin",fMin];

    rA=addParam[rA,nameCol,valCol," ", " "];
    rA=addParam[rA,nameCol,valCol,"rFitData",grFitDataName];

rA=addParam[rA,nameCol,valCol,"krFirstFull",grkFirstFull];
    rA=addParam[rA,nameCol,valCol,"krLastFull",grkLastFull];
    rA=addParam[rA,nameCol,valCol,"grSDFull",grSDFull];
    rA=addParam[rA,nameCol,valCol,"grZeroFull",grZeroFull];

rA=addParam[rA,nameCol,valCol,"grZeroErrFull",grZeroErrFull
];
    rA=addParam[rA,nameCol,valCol,"grInfFull",grInfFull];

```

```

rA=addParam[rA,nameCol,valCol,"grInfErrFull",grInfErrFull];
rA=addParam[rA,nameCol,valCol,"grTdFull",grTdFull];

rA=addParam[rA,nameCol,valCol,"grTdErrFull",grTdErrFull];

rA=addParam[rA,nameCol,valCol," "," "];
rA=addParam[rA,nameCol,valCol,"rFitData",grFitDataName];
rA=addParam[rA,nameCol,valCol,"krFirst500",grkFirst500];
rA=addParam[rA,nameCol,valCol,"krLast500",grkLast500];
rA=addParam[rA,nameCol,valCol,"grSD500",grSD500];

rA=addParam[rA,nameCol,valCol,"grZero500",grZero500];rA=add
Param[rA,nameCol,valCol,"grZeroErr500",grZeroErr500];rA=add
Param[rA,nameCol,valCol,"grInf500",grInf500];rA=addParam[rA
,nameCol,valCol,"grInfErr500",grInfErr500];
rA=addParam[rA,nameCol,valCol,"grTd500",grTd500];
rA=addParam[rA,nameCol,valCol,"grTdErr500",grTdErr500];

rA=addParam[rA,nameCol,valCol," "," "];
rA=addParam[rA,nameCol,valCol,"rFitData",grFitDataName];
rA=addParam[rA,nameCol,valCol,"krFirst200",grkFirst200];
rA=addParam[rA,nameCol,valCol,"krLast200",grkLast200];
rA=addParam[rA,nameCol,valCol,"grSD200",grSD200];

rA=addParam[rA,nameCol,valCol,"grZero200",grZero200];rA=add
Param[rA,nameCol,valCol,"grZeroErr200",grZeroErr200];rA=add
Param[rA,nameCol,valCol,"grInf200",grInf200];

rA=addParam[rA,nameCol,valCol,"grInfErr200",grInfErr200];
rA=addParam[rA,nameCol,valCol,"grTd200",grTd200];
rA=addParam[rA,nameCol,valCol,"grTdErr200",grTdErr200];

rA=addParam[rA,nameCol,valCol," "," "];
rA=addParam[rA,nameCol,valCol,"rFitData",grFitDataName];
rA=addParam[rA,nameCol,valCol,"krFirst100",grkFirst100];
rA=addParam[rA,nameCol,valCol,"krLast100",grkLast100];
rA=addParam[rA,nameCol,valCol,"grSD100",grSD100];
rA=addParam[rA,nameCol,valCol,"grZero100",grZero100];

rA=addParam[rA,nameCol,valCol,"grZeroErr100",grZeroErr100];
rA=addParam[rA,nameCol,valCol,"grInf100",grInf100];

rA=addParam[rA,nameCol,valCol,"grInfErr100",grInfErr100];
rA=addParam[rA,nameCol,valCol,"grTd100",grTd100];
rA=addParam[rA,nameCol,valCol,"grTdErr100",grTdErr100];

```

```

rA=addParam[rA,nameCol,valCol," "," "];
rA=addParam[rA,nameCol,valCol,"sFitData",gsFitDataName];

rA=addParam[rA,nameCol,valCol,"ksFirstFull",gskFirstFull];
rA=addParam[rA,nameCol,valCol,"ksLastFull",gskLastFull];
rA=addParam[rA,nameCol,valCol,"gsSDFull",gsSDFull];
rA=addParam[rA,nameCol,valCol,"gsZeroFull",gsZeroFull];

rA=addParam[rA,nameCol,valCol,"gsZeroErrFull",gsZeroErrFull];
];
rA=addParam[rA,nameCol,valCol,"gsInfFull",gsInfFull];

rA=addParam[rA,nameCol,valCol,"gsInfErrFull",gsInfErrFull];
rA=addParam[rA,nameCol,valCol,"gsTdFull",gsTdFull];

rA=addParam[rA,nameCol,valCol,"gsTdErrFull",gsTdErrFull];

rA=addParam[rA,nameCol,valCol," "," "];
rA=addParam[rA,nameCol,valCol,"sFitData",gsFitDataName];
rA=addParam[rA,nameCol,valCol,"ksFirst500",gskFirst500];
rA=addParam[rA,nameCol,valCol,"ksLast500",gskLast500];
rA=addParam[rA,nameCol,valCol,"gsSD500",gsSD500];
rA=addParam[rA,nameCol,valCol,"gsZero500",gsZero500];

rA=addParam[rA,nameCol,valCol,"gsZeroErr500",gsZeroErr500];
rA=addParam[rA,nameCol,valCol,"gsInf500",gsInf500];

rA=addParam[rA,nameCol,valCol,"gsInfErr500",gsInfErr500];
rA=addParam[rA,nameCol,valCol,"gsTd500",gsTd500];
rA=addParam[rA,nameCol,valCol,"gsTdErr500",gsTdErr500];

rA=addParam[rA,nameCol,valCol," "," "];
rA=addParam[rA,nameCol,valCol,"sFitData",gsFitDataName];
rA=addParam[rA,nameCol,valCol,"ksFirst200",gskFirst200];
rA=addParam[rA,nameCol,valCol,"ksLast200",gskLast200];
rA=addParam[rA,nameCol,valCol,"gsSD200",gsSD200];
rA=addParam[rA,nameCol,valCol,"gsZero200",gsZero200];

rA=addParam[rA,nameCol,valCol,"gsZeroErr200",gsZeroErr200];
rA=addParam[rA,nameCol,valCol,"gsInf200",gsInf200];

rA=addParam[rA,nameCol,valCol,"gsInfErr200",gsInfErr200];
rA=addParam[rA,nameCol,valCol,"gsTd200",gsTd200];
rA=addParam[rA,nameCol,valCol,"gsTdErr200",gsTdErr200];

rA=addParam[rA,nameCol,valCol," "," "];
rA=addParam[rA,nameCol,valCol,"sFitData",gsFitDataName];

```

```

rA=addParam[rA,nameCol,valCol,"ksFirst100",gskFirst100];
rA=addParam[rA,nameCol,valCol,"ksLast100",gskLast100];
rA=addParam[rA,nameCol,valCol,"gsSD100",gsSD100];
rA=addParam[rA,nameCol,valCol,"gsZero100",gsZero100];

rA=addParam[rA,nameCol,valCol,"gsZeroErr100",gsZeroErr100];
rA=addParam[rA,nameCol,valCol,"gsInf100",gsInf100];

rA=addParam[rA,nameCol,valCol,"gsInfErr100",gsInfErr100];
rA=addParam[rA,nameCol,valCol,"gsTd100",gsTd100];
rA=addParam[rA,nameCol,valCol,"gsTdErr100",gsTdErr100];

rA=addParam[rA,nameCol,valCol," "," "];

rA=addParam[rA,nameCol,valCol,"grrFluct12[[2]]",grrFluct12[
[2]]];

rA=addParam[rA,nameCol,valCol,"grrFluct12[[3]]",grrFluct12[
[3]]];

rA=addParam[rA,nameCol,valCol,"grrFluct12[[4]]",grrFluct12[
[4]]];
rA=addParam[rA,nameCol,valCol,"gZero3Pts",gZero3Pts];

rA=addParam[rA,nameCol,valCol,"gZeroErr3Pts",gZeroErr3Pts];
rA=addParam[rA,nameCol,valCol,"gTd3Pts",gTd3Pts];
rA=addParam[rA,nameCol,valCol,"gTdErr3Pts",gTdErr3Pts];
rA=addParam[rA,nameCol,valCol,"gSD3Pts",gSD3Pts];

rA=addParam[rA,nameCol,valCol," "," "
"];rA=addParam[rA,nameCol,valCol,"model",model];

(*rA//MatrixForm*)
(*Print[Length[rA]];
Print[rA[[35]]];
Abort[];*)

Do>(*i down*)
jLast=Length[rA[[i]]];
Do>(*j across*)
WriteString[res,ToString[ff[rA[[i,j]]]]];

If[j!=jLast,WriteString[res,","],WriteString[res,"\n"]]
,{j,1,jLast}
]
,{i,Length[rA]}
];

```

```

Close[res];
Print["results filename= ",resultsFileName];
restIPrint[]
);
(*saveResults[];*)

(*****
*****
(* combine all graphics and export
*)
(*****
*****
ClearAll[savePlots];
savePlots[]:=
(
saveIPrint[0];
ClearAll[pageLabel];

(*bs(*BaseStyle*)={FontFamily[]"CourierNew",FontSize[]12};*)
pageLabel=
Framed[
Graphics[
Inset[
Pane[
TextCell[
StringJoin[inFileName," =
inFileName","\n",ToString[ iDot ] ," = iDot" ,
"\n",FileBaseName[NotebookFileName[]]," =
notebook","\n",dateString," = date","\n", timeString, " =
time"],
"Text",
FontSize->10
],
144 (*pane width pts*)
],
{0,0} (*inset pos*)
],
ImageSize->{144,108}
]
];
allPlots=GraphicsGrid[{
{pageLabel,vhEffPlot,bkgEffPlot, bkgCorrPlot},
{vchcPlot,vchcDiffPlot,sPlot,rPlot},
{gsrFullPlot, gsr500Plot ,gsr200Plot, gsr100Plot},
{grsFullPlot,grs500Plot,grs200Plot,grs100Plot},
{gsFitPlotFull,gsFitPlot500, gsFitPlot200,

```

```

gsFitPlot100},
    {grFitPlotFull,grFitPlot500, grFitPlot200,
grFitPlot100}
    }, ImageMargins->{{36,36},{36,36}}];
    plotFileName=base<>"-"<>"plots"<>"-
"<>timeStampString<>".pdf";
    Export [plotFileName, allPlots];
    Print["plot filename= ",plotFileName];
    restIPrint[];
    );
(*savePlots[]*)

(*****
*****
(* show analysis done this dot
*)
(*****
*****
ClearAll[done];
done[]:=Print["dot ", iDot, " processed and products
saved\n\n"];
(*done[]*)

(*****
*****
(* process one dot completely
*)
(*****
*****
ClearAll[processOneDot];
processOneDot[]:=
    (optimize[];
    showOptResults[];
    fitAnisDecay[];
    fitAnisDecay3Pts[];
    fitIntenDecay[];
    saveResults[];
    savePlots[];
    done[]);
(*processOneDot[]*)

(*****
**)
(*
    process dots this file
*)
(*
    iDot=-1 □ no background traces
*)

```

```

(*****
**)
ClearAll[dotsThisFile];
dotsThisFile[(*inFileName_,iFirstDot_,iLastDot_*)]:=Module[
{(*inFileName*)(*,iFirstRow,isBkgs,data,dim,iLastRow,nCols,
nDots,n,iDot,cBkg,iTime,iv,ih,ibv,ibh,ibTime,v,h,bv,bh,bvf,
bhf,gbv,gbh,vhRawPlot,bkgRawPlot,
bkgCorrPlot,inputPlot,s,dep*)},
saveIPrint[];

(* generate timestamp for all output this pass *)
isStamp=True;
tdList=DateList[];
dateFormat={"Year","-","Month","-","Day"};
timeFormat={"Hour",":","Minute",":","Second"};
dateString=DateString[tdList,dateFormat];
timeString=DateString[tdList,timeFormat];
timeStampFormat={(*"-",*)"Year",(*"-",*)"Month",(*"-",
*,*)"Day",*"-","Hour24","Minute","Second"};

timeStampString=If[isStamp,DateString[tdList,timeStampForma
t],""];
(*dP[timeStampString];*)

(*inFileName="IgE-1-LRB.csv";
q=DialogInput[{}],Column[

{"inFileName=\t"InputField[Dynamic[inFileName],String,Align
ment☐Right],

(*"isBkgs=\t"InputField[Dynamic[isBkgs],Number,Alignment☐Ri
ght],

"CBkgs=\t"InputField[Dynamic[cBkg],Number,Alignment☐Right],

"iFirst=\t"InputField[Dynamic[iFirstRow],Number,Alignment☐R
ight],*)
Button["proceed",DialogReturn[{inFileName,
isBkgs,cBkg,iFirst}]]]];
inFileName=q[[1]];*)(* dP[inFileName];*)

data=Import[inFileName];(*dP[inFileName];*)
dim=Dimensions[data]; (*dP[dim];*)
len=dim[[1]];(*dP[len];*)
nCols=dim[[2]]; (*dP[nCols];*)
If[Mod[nCols,3]!=0, Print["nCols not divisible by
3"];Exit[]];

```

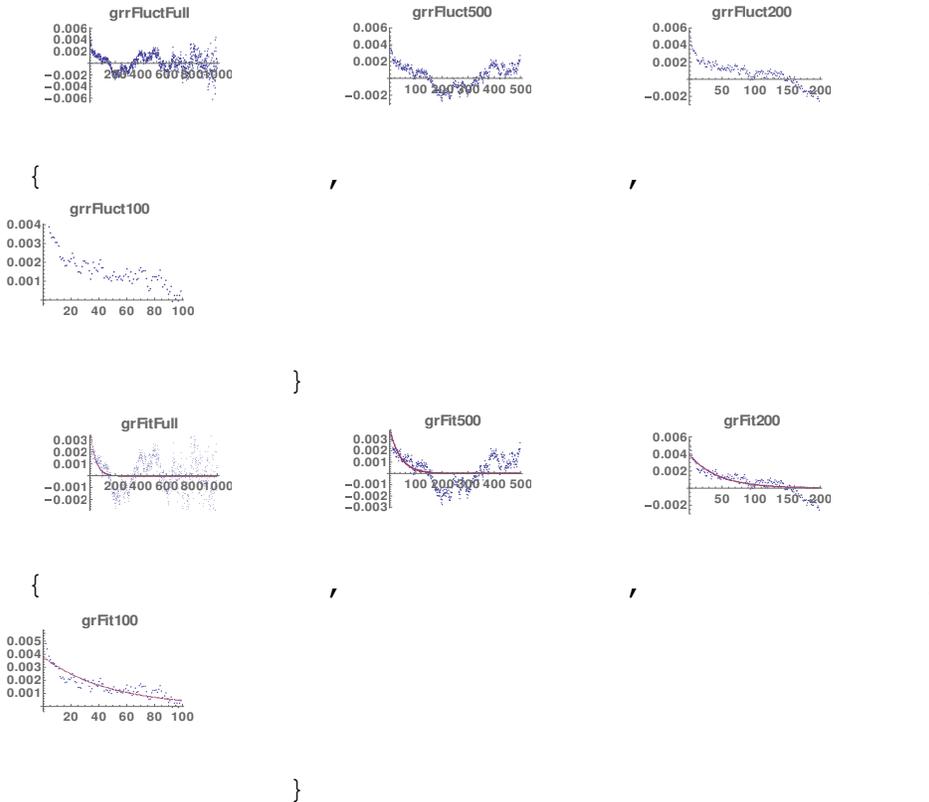
```

nDots=(nCols-6)/3; (*dP[nDots];*)
(*nDots=1;*)
n(*num pts*)=len-iFirst +1; (*dP[n];*)
i1=If[iFirstDot<1, 1,iFirstDot];
i2=If[iLastDot<1, nDots,iLastDot];

For [iDot=i1,iDot<=i2,iDot++,
  (*Print info this dot*)
  Print["Analyzing ",inFileName," dot=", iDot, " of ",
nDots];
  Print["Model: ",model];
  base= StringTake[inFileName, StringLength[inFileName]-
4]<>If[iDot!=-1,"-d"<>IntegerString[iDot,10,2],""];
  ibTime=1;ibv=2;ibh=3;
  iTime=4+3*iDot;iv=5+3*iDot;ih=6+3*iDot;
  dataRange={};
  time=data[[iFirst;;len,iTime]]; (*1st dim is DOWN, 2nd
ACROSS *)
  vEff=data[[iFirst;;len,iv]]; (*starts 2nd row,
indicated col *)
  hEff=data[[iFirst;;len,ih]]; (*starts 2nd row,
indicated col *)
  If [iDot!=-1, (* yes, bkgS *)
    bvEff=data[[iFirst;;len,ibv]]; (*starts 2nd row,
indicated col *)
    bhEff=data[[iFirst;;len,ibh]],
    (*else*)
    bvEff=Table[0,{n}];
    bhEff=Table[0,{n}]
  ];(*if*)

  bvf=bvEff-Mean[bvEff]; bhf=bhEff-Mean[bhEff];
  gbv=dataCorrelate[bvf,bvf];
  gbh=dataCorrelate[bhf,bhf];
  range={0, 1.25Max[vEff,hEff]};
  inputPlots={
    vhEffPlot=ListPlot[{vEff,hEff},PlotLabel->
"vhEff",PlotRange->range],
    bkgEffPlot=ListPlot[{bvEff,bhEff},PlotLabel->
"bkgSRaw",PlotRange->
range],bkgCorrPlot=ListPlot[{gbv,gbh},PlotLabel->
"bkgCorr"]];
  ListPlot[{gbv,gbh},PlotLabel->"bkgCorr"];
  processOneDot[]
  ] ;(*for*)
restIPrint[];(* ";" needed here; otherwise prints "0"*)
]; (*module*)

```

```

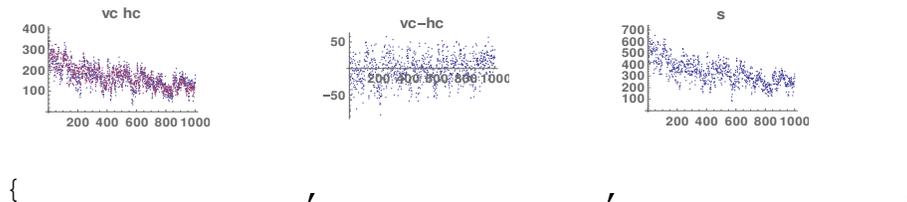
results filename= 2013-01-14-IgE-4-LRB-d02-results-
20200728-150405.csv
plot filename= 2013-01-14-IgE-4-LRB-d02-plots-20200728-
150405.pdf
dot 2 processed and products saved

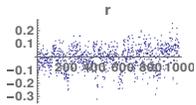
```

```

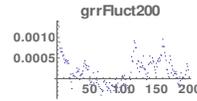
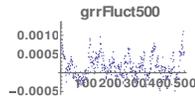
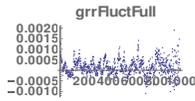
Analyzing 2013-01-14-IgE-4-LRB.csv dot= 3 of 11
Model: v120-NewGC-wtdpnc
Optimization successful! - p = 7.91313*10-8
iPass=11 iDot= 3 of 11
RAvg=0 RSlope=5.58685 RCurve=0 varDiff=0 zeroErr=0
mlpNeg=0 mlpRSDistCorr=19.0368 mlpRSPmCorr=3.09615
mlpRSWpmCorr=7.91313e-8
p=7.91313e-8 g=0.350105 b=1.91124 c=-26.1886 f=300

```

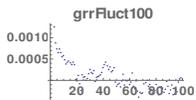




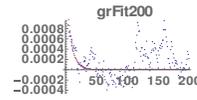
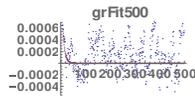
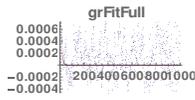
}



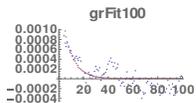
{



}



{



}

```

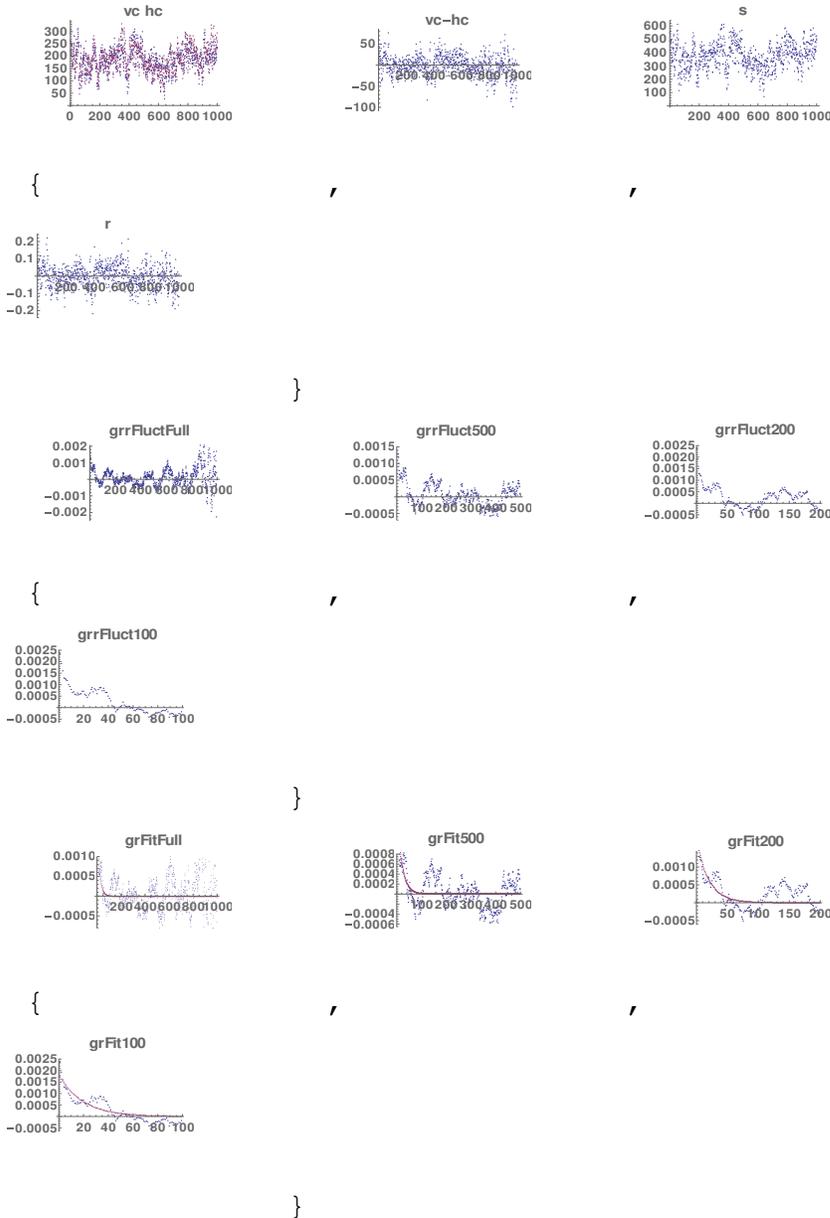
NonlinearModelFit::cvmit: Failed to converge to the
requested accuracy or precision within 100 iterations.
results filename= 2013-01-14-IgE-4-LRB-d03-results-
20200728-150411.csv
plot filename= 2013-01-14-IgE-4-LRB-d03-plots-20200728-
150411.pdf
dot 3 processed and products saved

```

```

Analyzing 2013-01-14-IgE-4-LRB.csv dot= 4 of 11
Model: v120-NewGC-wtdpmc
Optimization successful! - p = 2.75128*10-6
iPass=11 iDot= 4 of 11
RAvg=0 RSlope=10.0824 RCurve=0 varDiff=0 zeroErr=0
mlpNeg=0 mlpRSDistCorr=4.03122 mlpRSPmCorr=1.12456
mlpRSWpmCorr=2.75128e-6
p=2.75128e-6 g=0.459352 b=2.09034 c=-12.734 f=300

```



```

}
results filename= 2013-01-14-IgE-4-LRB-d04-results-
20200728-150417.csv
plot filename= 2013-01-14-IgE-4-LRB-d04-plots-20200728-
150417.pdf
dot 4 processed and products saved

```

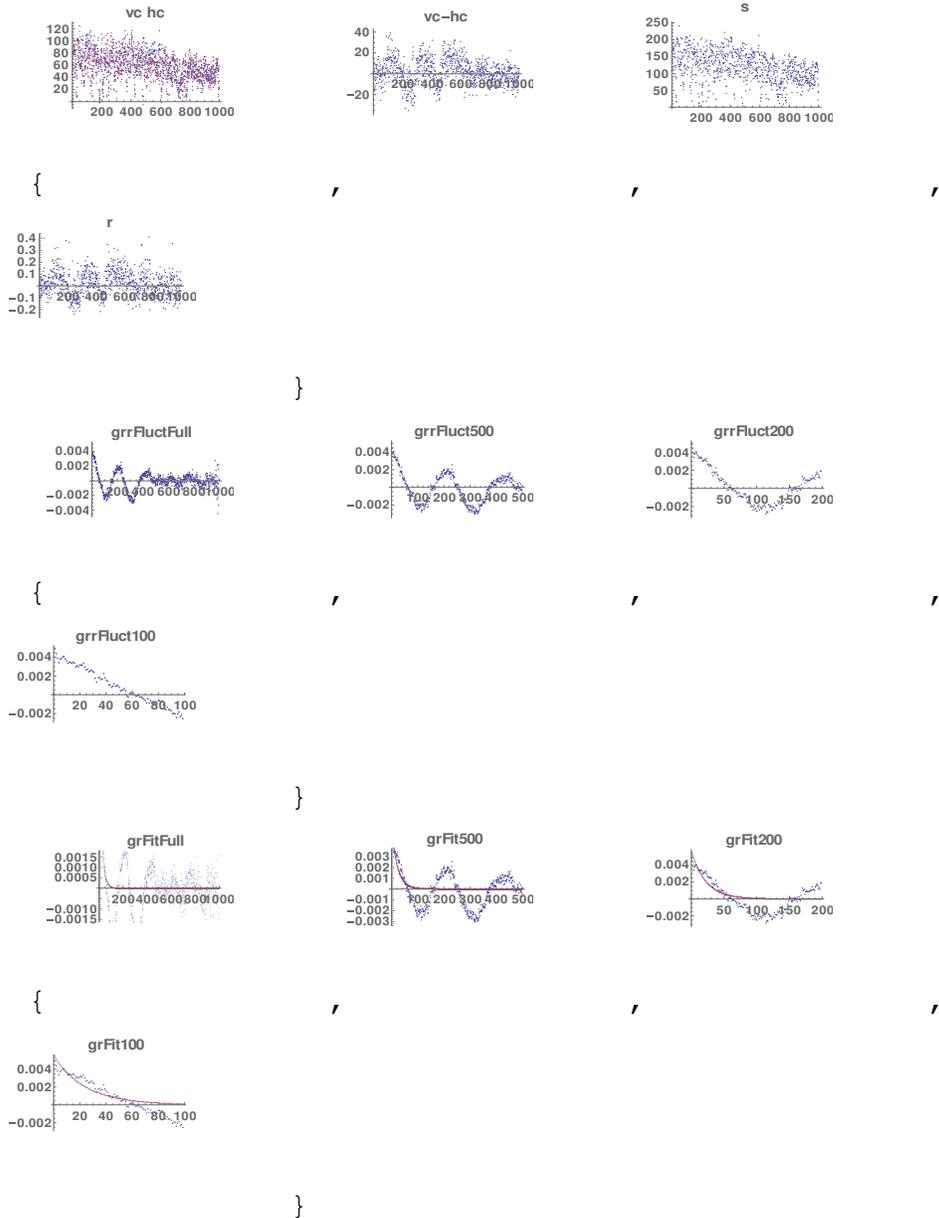
```

Analyzing 2013-01-14-IgE-4-LRB.csv dot= 5 of 11
Model: v120-NewGC-wtdpmc
Optimization successful! - p = 0.0000151239
iPass=11 iDot= 5 of 11
RAvg=0 RSlope=10.8477 RCurve=0 varDiff=0 zeroErr=0
mlpNeg=2.24106 mlpRSDistCorr=21.7861 mlpRSPmCorr=27.7224

```

mlpRSWpmCorr=0.0000151239

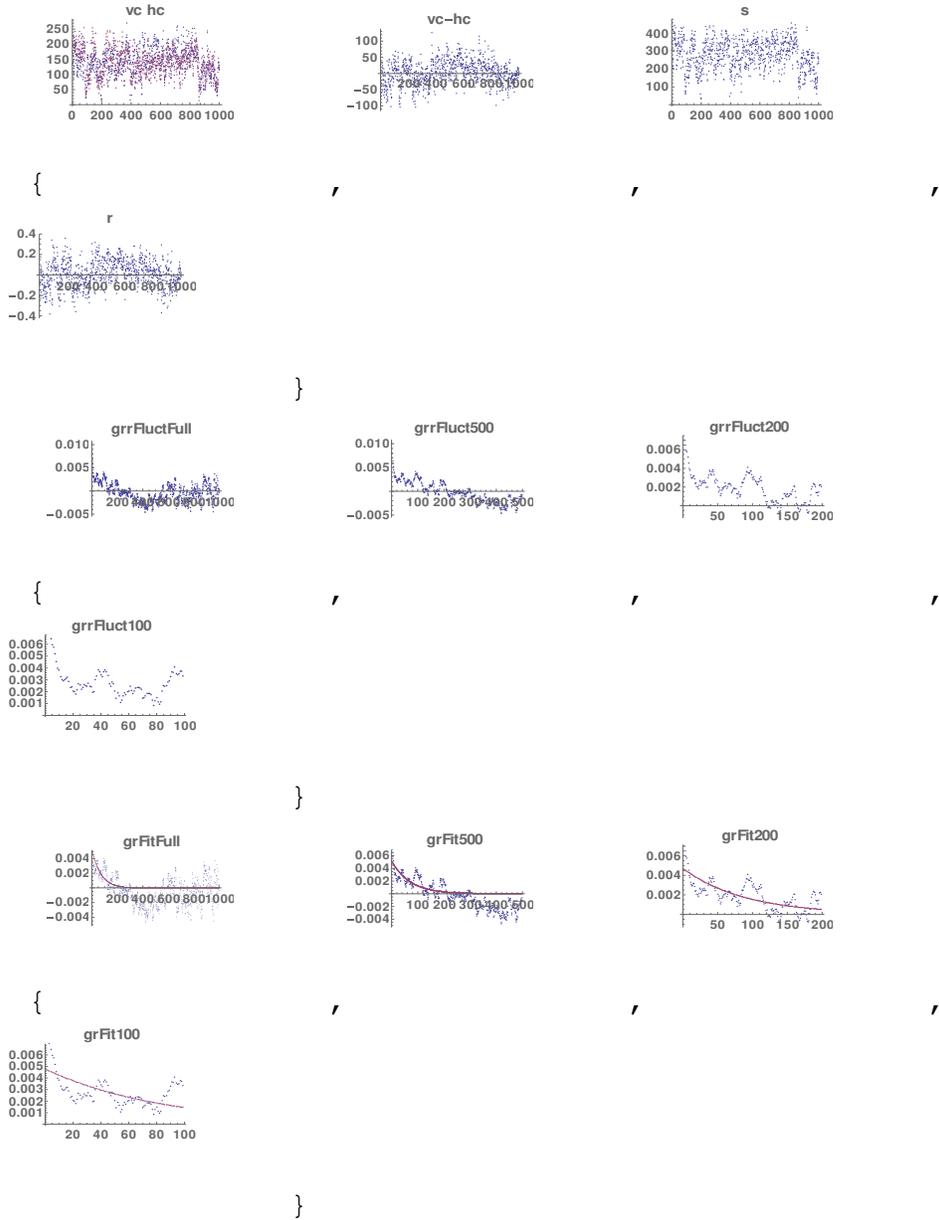
p=0.0000151239 g=0.297798 b=-2.28086 c=-3.29489 f=300



results filename= 2013-01-14-IgE-4-LRB-d05-results-20200728-150423.csv
plot filename= 2013-01-14-IgE-4-LRB-d05-plots-20200728-150423.pdf
dot 5 processed and products saved

Analyzing 2013-01-14-IgE-5-LRB.csv dot= 2 of 13
Model: v120-NewGC-wtdpmc
Optimization successful! - p = 0.00827842
iPass=1 iDot= 2 of 13

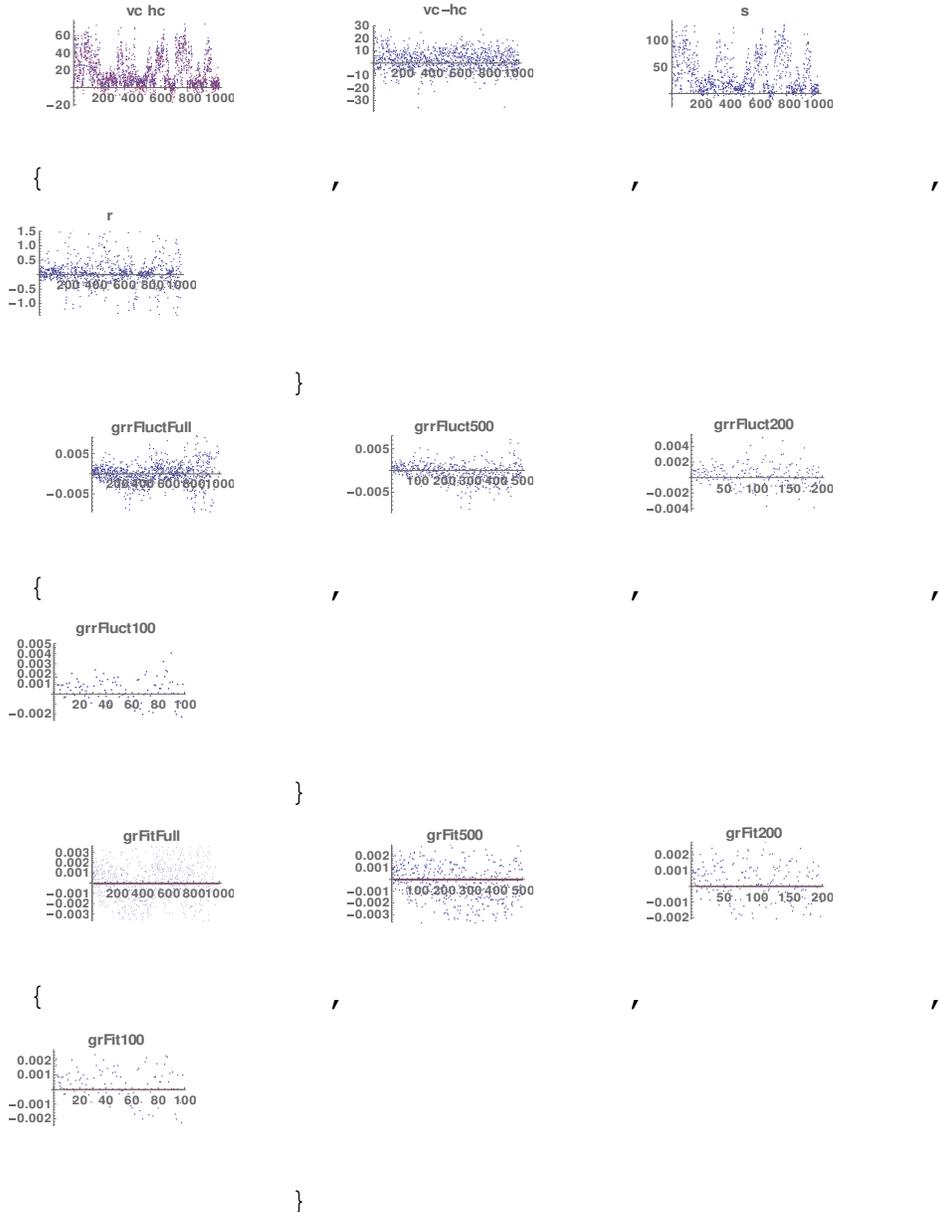
RAvg=0 RSlope=0.686262 RCurve=0 varDiff=0 zeroErr=0
 mlpNeg=0 mlpRSDistCorr=1.62927 mlpRSPmCorr=0.722485
 mlpRSWpmCorr=0.124673
 p=0.124673 g=0.437762 b=-2.28086 c=-2.58008 f=300



results filename= 2013-01-14-IgE-5-LRB-d02-results-
 20200728-150428.csv
 plot filename= 2013-01-14-IgE-5-LRB-d02-plots-20200728-
 150428.pdf
 dot 2 processed and products saved

Analyzing 2013-01-14-IgE-5-LRB.csv dot= 3 of 13
 Model: v120-NewGC-wtdpmc

Optimization successful! - p = 1.96067*10⁻⁶
 iPass=18 iDot= 3 of 13
 RAvg=0 RSlope=7.68376 RCurve=0 varDiff=0 zeroErr=0
 mlpNeg=195.729 mlpRSDistCorr=63.6675
 mlpRSPmCorr=0.0974481 mlpRSWpmCorr=1.96067e-6
 p=1.96067e-6 g=0.479288 b=-2.46901 c=3.63387 f=300

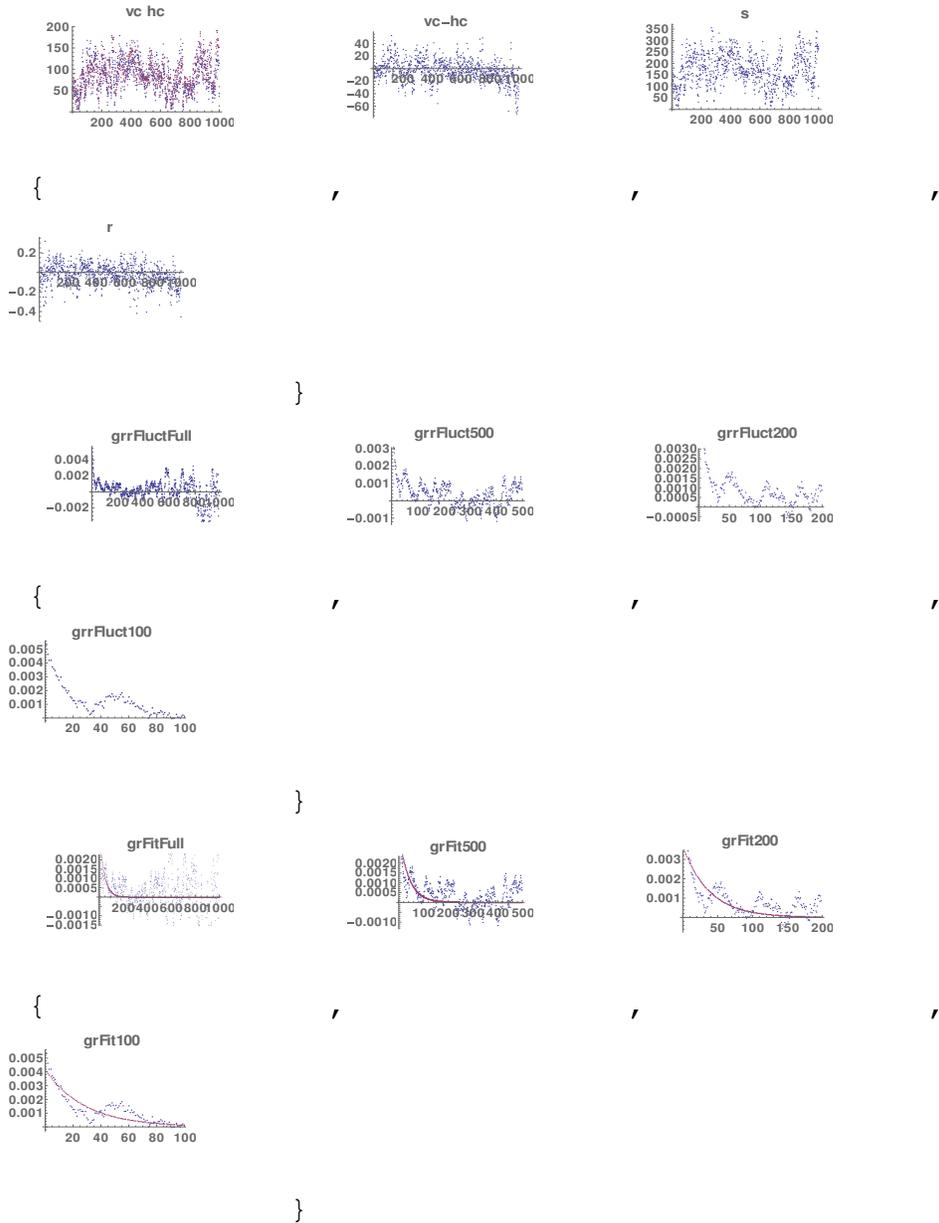


results filename= 2013-01-14-IgE-5-LRB-d03-results-20200728-150435.csv
 plot filename= 2013-01-14-IgE-5-LRB-d03-plots-20200728-150435.pdf
 dot 3 processed and products saved

```

Analyzing 2013-01-14-IgE-5-LRB.csv dot= 4 of 13
Model: v120-NewGC-wtdpmc
Optimization successful! - p = 0.0000251235
iPass=11 iDot= 4 of 13
RAvg=0 RSlope=15.2808 RCurve=0 varDiff=0 zeroErr=0
mlpNeg=0 mlpRSDistCorr=19.653 mlpRSPmCorr=15.9208
mlpRSWpmCorr=0.0000251235
p=0.0000251235 g=0.289786 b=3.17398 c=-6.38602 f=300

```

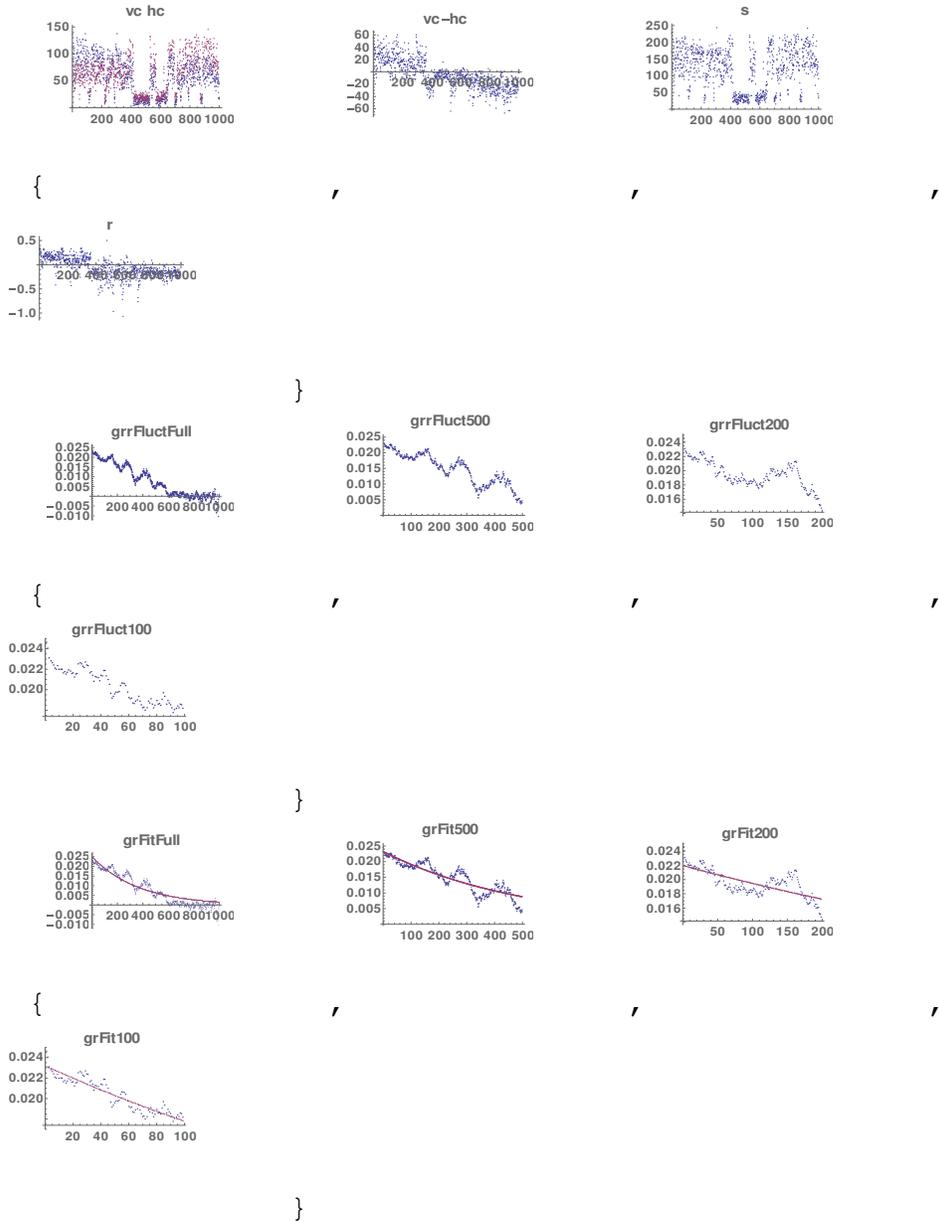


```

results filename= 2013-01-14-IgE-5-LRB-d04-results-
20200728-150441.csv
plot filename= 2013-01-14-IgE-5-LRB-d04-plots-20200728-
150441.pdf
dot 4 processed and products saved

```

Analyzing 2013-01-14-IgE-5-LRB.csv dot= 5 of 13
 Model: v120-NewGC-wtdpmc
 Optimization successful! - p = 0.00184716
 iPass=11 iDot= 5 of 13
 RAvG=0 RSlope=17.3646 RCurve=0 varDiff=0 zeroErr=0
 mlpNeg=0.0135595 mlpRSDistCorr=49.0963 mlpRSPmCorr=53.747
 mlpRSWpmCorr=0.00184716
 p=0.00184716 g=0.262132 b=4.80445 c=-11.0218 f=300



results filename= 2013-01-14-IgE-5-LRB-d05-results-20200728-150447.csv
 plot filename= 2013-01-14-IgE-5-LRB-d05-plots-20200728-

150447.pdf

dot 5 processed and products saved

Program Code 3: Image Corr v.140

```
(* GOAL: assign b, c and g-factor to QD traces *)
(* so as to minimize dependence of r and s *)
(* use the new defn of g as 1+f and 1-f *)
(*****
)
```

```

(*****
)
(* initialize Mathematica system
*)
(*****
)
ClearSystemCache[];
SetDirectory[NotebookDirectory[]];
Needs["CCompilerDriver`"];
Needs["SymbolicC`"];
Needs["CompiledFunctionTools`"];
Unprotect[CompiledFunctionTools`Private`getInstruction];
CompiledFunctionTools`Private`getInstruction[line_, {0, _}] :=
CompiledFunctionTools`Private`getInstruction[line, {3, 1}]
SetDirectory[NotebookDirectory[]];
(*SetOptions[$FrontEndSession, PrintingStyleEnvironment <math>\square</math>"Working"];*)
Off[General::munfl];
{\pard{}}[FittedModel::precw];
Off[NonlinearModelFit::sszero];
Off[FittedModel::constr];

ClearAll[gPrint,
one, remOdd, dP, aP, se, restIPrint, saveIPrint];
gPrint= If[iPrint!=0, Print[##]]&;
iPrint=1;
iPrintS=1;
one=#1&;
remOdd[e_, x_] := ((e/.{x->Power[x,2]})/.{ x2->0, x6->0, x10->0, x14-> 0})/.{x-> x1/2} (*remove odd pwr of x*)
SetAttributes[dP, HoldAll];
dP[z_] := If[(iPrint!=0 && ValueQ[z]==True) ||
(ValueQ[iPrint]==False), Print[HoldForm[z], "=", ReleaseHold[z]
](*//MatrixForm*)]]; (* debug print showing variable name*)
aP[a_] := Module[{last}, last=Min[8,
Length[a]]; Evaluate[dP[a[[1;;last]]]]; (*array print
showing 8 list members max*)
se[x_] := Simplify[Expand[x]]; (* simplify-expand *)
restIPrint[] := (iPrint=iPrintS);
saveIPrint[] := (iPrintS=iPrint);

(*****
**)
(* global parameters
*)
(*****
**)

```

```

(*inFileName="IgE-1-LRB.csv";*)
iFirstDot=1; iLastDot=1; (*"0" means all dots*)
iDot=1; (* used when NOT entering via "dataImport"*)
iPrint=1;
cBkg=300.0(*319.75*); (* initial value of camera bkg *)
isStamp=True;
iWt=2;(*2=Gaussian; 1=Poisson*)
iGeom=1;(*1=polariz, 2=anis*)
iPrint=1; (* print all dP instances *)
pGoal=0.01;
maxPass=2000;
tauMin=2-6;
g0=0.4;b0=0.0;c0=0.0;f0=-1.; (*"-1" means use prog's camera
bkg *)
iFirst=2;iLast=1001; kFirst=0; kLast=0;
nBkgs=4;
isBkgs=1;
isSim=0; (* simulation? *)
iWeight=2; (* 1=uniform wt, 2=s2 wt; 3=full s,r pt wt*)
nParams=2; (*2 or 3 p *)
isSavePlots=True;

(*****
*)
(*****
*)
(* UTILITIES *)
(*****
*)
(*****
*)

(*****
)
(* set graphics point sizes (fract of total graph)
*)
(*****
)
SetOptions[{Plot,ListPlot, ListLinePlot},
  PlotStyle->PointSize[0.01],
  PlotStyle->{RGBColor[0.2472,0.24,0.6],
    RGBColor[0.6,0.24,0.4429],
    RGBColor[0.6,0.5470,0.24],
    RGBColor[0.24,0.6,0.3369]},
  BaseStyle->{
    FontFamily->"Helvetica",
    FontWeight->"Bold" },

```

```

ImageSize->{144,108} (*Medium*)
];

(*****
*****
(*  define pattern replace to pull constants out of sums
*)
(*****
*****
(* patt /; test is a pattern which matches only if the
evaluation of test yields True *)
(* lhs  $\square$  rhs "\\\"\\\"RuleDelayed\\\"\\\"\\\"\\\" represents a
rule that transforms lhs to rhs evaluating rhs only after
the rule is used *)
(* FreeQ[expr,form] yields True if no subexpression in expr
matchesformand yields False *)
ClearAll[BringOut,outrules];
outrules={Sum[f_+
g_,it:{x_Symbol,___}]>Sum[f,it]+Sum[g,it],Sum[c_
f_,it:{x_Symbol,___}]>c
Sum[f,it]/;FreeQ[c,x],Sum[c_,it:{x_Symbol,___}]>c
Sum[1,it]/;FreeQ[c,x]];
BringOut[s_]:=s //. outrules
(*BringOut [Sum [c*i*x[[i]]1,{i,n}]];*

(*****
*****
(*  map Sum function over sum of terms
*)
(*****
*****
(*ClearAll[x,y,z,qd0,n];
qd0= x[[i]] +2y[[i]]^2+ 3 z[[i]]^3;
gd0=Total[Sum[#, {i,1,n}]&/@(List@@qd0)];
BringOut[gd0]*)

(*****
*****
(*  abbreviate FortranForm for printing
*)
(*****
*****
ClearAll[ff];
ff[x_]:=FortranForm[x];

(*****
*****

```

```

(*  define gRound to get suitable form printing large,
small nums  *)
(*****
*****
ClearAll[gR];
gR[x0_,sf_]:= (* "g Round" *)Module[{e,xr,x},
  (x=Re[x0];e=MantissaExponent[x][[2]]-sf;
  xr=(10^e)*Round[x*10^(-e)];
  xr)];

(*****
*****
(*  define gF "George Format" for compact 1-line output
*)
(*****
*****
ClearAll[gF];
gF[x_]:=ff[gR[x,6]];

(*****
*)
(*****
*)
(*  FUNCTIONS THIS PROGRAM  *)
(*****
*)
(*****
*)

(*****
)
(*          SZEKELY dist covariance via Hu's algorithm
*)
(*****
)
ClearAll[distCovCompile];
distCovCompile=Compile[
  {{xIn,_Real,1},
  {yIn,_Real,1}},

Module[{index,si,sLast,t,ax,v,nw,zeros,idx,iv1,iv2,iv3,iv4,

i,r,s,gap,k,idxr,csumv,idx1,idx2,st1,st2,e1,e2,kf,covterm,c
1,c2,c3,c4,d,ySorted,by,nsq,ncb,nq,term1,term2,term3,x,y,n,
j},
  (*iPrint=0;*)
  n=Length[xIn];

```

```

index=Ordering[xIn];
x=Sort[xIn]; (*gPrint["xSorted=",x//MatrixForm];*)
y=yIn[[index]]; (*gPrint["yReordered=",y//MatrixForm];*)
si=Accumulate[x]; (*gPrint["si=",si//MatrixForm];*)
sLast=si[[n]];
t=Table[i, {i, -(n-
2), n, 2}]; (*gPrint["t=",t//MatrixForm];*)
ax=t*x+(sLast-2 si); (*gPrint["ax=",ax//MatrixForm];*)
v={x,y,x*y}[]; (*gPrint["v=",v//MatrixForm];*)
nw=Dimensions[v][[2]];
zeros=Table[0,n];

idx={Table[i,{i,n}],zeros}[]; (*gPrint["idx=",idx//MatrixForm
];*)
iv1=Table[0.,n];iv2=iv1;iv3=iv1;iv4=iv1;
i=1;r=1;s=2;

(*While [i<n,*)
Label[startWhileI];
If[i>=n, Goto[endWhileI]];
gap=2*i;
k=0;
idxr=idx[[All,r]]; (*gPrint["idxr=",idxr//MatrixForm];*)
csumv=Prepend[Accumulate[v[[idxr]]],Table[0,nw]];
(* OK to here*)
(*gPrint["csumv=", csumv//MatrixForm];*)

(*For [j=1, j<n, j=j+gap,*)
j=1;
Label[startForJ];
If [j>=n, Goto[endForJ]];
(*st1=j;e1=Min[st1+i-1,n];st2=j+i;e2=Min[st2+i-1,n];*)
st1=j;e1=st1+i-1; If[e1>=n,e1=n];st2=j+i;e2=st2+i-
1;If[e2>=n,e2=n];
(*While [(st1<=e1)&&(st2<=e2),*)
Label[startWhileSt];
If[(st1>e1)|| (st2>e2),Goto[endWhileSt]];
k=k+1;(*gPrint ["i=",i," j=",j, " k=",k, " st1=",st1,"
e1=",e1," st2=",st2," e2=",e2];*)
idx1=idxr[[st1]];idx2=idxr[[st2]];
(*gPrint ["idx1=",idx1," idx2=",idx2];*)
If[ y[[idx1]]>=y[[idx2]],
idx[[k,s]]=idx1;
st1=st1+1,
(*else*)
idx[[k,s]]=idx2;
st2=st2+1;

```

```

        iv1[[idx2]]      =iv1[[idx2]]+e1-st1+1;iv2[[idx2]]
=iv2[[idx2]]+(csumv[[e1+1,1]]-csumv[[st1,1]]);

        iv3[[idx2]]      =iv3[[idx2]]+(csumv[[e1+1,2]]-
csumv[[st1,2]]);
        iv4[[idx2]]      =iv4[[idx2]]+(csumv[[e1+1,3]]-
csumv[[st1,3]]);
    ];(*end If[y[idx1...]*)
    (*gPrint ["ivN=",{iv1,iv2,iv3,iv4}□//MatrixForm];*)

    (*); (*end While[(st1...)*]*)
    Goto[startWhileSt];
    Label[endWhileSt]; (*end While[(st1...)*]

    (*gPrint["i,j=",i," ",j];*)
    If[ st1<=e1,
        kf=k+e1-st1+1; (*gPrint["kf=",kf];*)
        idx[[ (k+1) ;; kf, s]]=idxr[[st1;;e1]];
        k=kf,
        (*else*)
        If[st2<=e2,
            kf=k+e2-st2+1;
            (*gPrint[idx[[ (k+1) ;; kf, s]]];*)
            idx[[ (k+1) ;; kf, s]]=idxr[[st2;;e2]];
            k=kf
        ];
    ];(*end If[st2...]*)

    (*gPrint["kf=",kf," idx=",idx//MatrixForm];*)
    (*gPrint["idx=",idx//MatrixForm];*)
    (*);(* end For[j=1...]*)
    j=j+gap;
    Goto[startForJ];
    Label[endForJ];(* end For[j=1...*)

    i=gap;
    r=3-r;s=3-s;
    (*);(* end While i<n*)*)
    Goto[startWhileI];
    Label[endWhileI];(* end While i<n*)
    covterm=n*(x-Mean[x]).(y-Mean[y]);

    c1=iv1.v[[All,3]];
    c2=Total[iv4];
    c3=iv2.y;
    c4=iv3.x;

```

```

d=4*((c1+c2)-(c3+c4))-2*covterm;
ySorted=y[[idx[[Table[i,{i,n,1,-1}],r]]]];
si=Accumulate[ySorted];
sLast=si[[n]];
by=Table[0.,n];
by[[idx[[n;;1;;-1,r]]]]=Table[i,{i,-(n-
2),n,2}]*ySorted+(sLast-
2*si);nsq=N[n*n];ncb=nsq*n;nq=ncb*n;term1=d/nsq;term2=2*(ax
.by)/ncb;term3=Total[ax]*Total[by]/nq;
Return[N[(term1+term3)-term2]
],(*end module/function*)
RuntimeAttributes->{Listable},Parallelization->False,
CompilationTarget->"C",RuntimeOptions->"Speed"
];(*end compile*)
<<CompiledFunctionTools`
(*CompilePrint[distCovCompile]*)
(*yIn=N[{3,5,7,3,8,4,6,7}];Print["yIn=",yIn];
xIn=N[{1,5,3,2,4,6,7,5}];Print["xIn=",xIn];
Print["distCov=",distCovCompile[xIn,yIn]];*)

(*****
*)
(* SZEKELY dist correlation from dist covariances
*)
(*****
*)
ClearAll[distCorrFunct];
distCorrFunct[x_,y_]:=Module[{covAB,covAA,covBB},
covAB=Evaluate[distCovCompile[x,y]];
covAA=Evaluate[distCovCompile[x,x]];
covBB=Evaluate[distCovCompile[y,y]];

(*Print["covAB=",covAB,"\ncovAA=",covAA,"\ncolBB=",covBB];*
)
Return[N[covAB/Sqrt[covAA*covBB]]];
];
(*yIn=N[{3,5,7,3,8,4,6,7}];Print["yIn=",yIn];
xIn=N[{1,5,3,2,4,6,7,5}];Print["xIn=",xIn];
Print["distCorr=",distCorrFunct[xIn,yIn]];*)

(*****
*****
*)
(* fast autocorrelation using LISTCORRELATE
*)
(*****
*****
)
ClearAll[dataCorrelate];

```

```

dataCorrelate[t_,u_]:=Module[{n},
z=ListCorrelate[t,u,1,0];n=Length[t];Table[z[[i]]/(n-
i+1),{i,n}]];
(*t={4,3,2,1}
u={1,2,3,4}
z=ListCorrelate[t,u,1,0]
z1=Table[z[[i]]/(Length[z]-i+1),{i,Length[z]}]
dataCorrelate[t,u]*)

(*****
*****
(*  weighted covariance function
*)
(*****
*****
ClearAll[wtdCov];
wtdCov[f_,g_,w_]:=Module[{n,wTot,temp},
n=Length[f];
wTot=Total[w];
temp=(wTot*Total[f*g*w]-Total[f*w]*Total[g*w])/wTot2
]; (* fails for f=g and i=1 *)

(*****
*****
(*  weighted correlation coeff
*)
(*****
*****
ClearAll[wtdCorrCoeff];
wtdCorrCoeff[f_,g_,w_]:=Module[{} ,
wtdCov[f,g,w]/\[Sqrt](wtdCov[f,f,w]*wtdCov[g,g,w])];

(*****
)
(*          faster version of brute force correlate
*)
(*****
)
SetSystemOptions["CompileOptions"-
>{"CompileReportExternal"->True}];
SetSystemOptions["CompileOptions"-
>{"CompileReportExternal"->False}];
ClearAll[trueCorr,i];
trueCorr=Compile[(* wx, wy are just 1/vx, 1/vy for each pt
*)

{{x,_Real,1},{y,_Real,1},{wx,_Real,1},{wy,_Real,1}},Module[

```

```

{n,nFact,tau,txy=0.0,tx=0.0,ty=0.0,
  tw=0.0,w=0.0,iy,ix,vx=0.0,vy=0.0,xVar,yVar,
  x0=0.0,y0=0.0,kList,gxy,gx,gy,gw},
n=Length[x];
gxy=Table[0.0,{n}];gx=gxy;gy=gxy;gw=gxy;
xVar=1.0/wx;yVar=1.0/wy;
kList=Table[n-tau+1,{tau,n}];
For[tau=1,tau<=n,tau++,
  txy=0.0;tx=0.0;ty=0.0;tw=0.0;
  For[ix=1,ix<=n-tau+1,ix++,
    iy=ix+tau-1;
    x0=x[[ix]];y0=y[[iy]];
    vx=xVar[[ix]];vy=yVar[[iy]];
    w=(vy*x0^2+vx y0^2+vx*vy)^-1;
    txy+=w*x0*y0;tx += w*x0;ty+=w*y0;tw+=w
  ];(*for i*)
  gxy[[tau]]=txy; (* weighted sums, not averages*)
  gx[[tau]]=tx;
  gy[[tau]]=ty;
  gw[[tau]]=tw
  ];(*for tau*)
Return[{gxy,gx,gy,gw}]
],(*end module/function*)
RuntimeAttributes->{Listable},
Parallelization->False,
CompilationTarget->"C",
RuntimeOptions->"Speed"
];(*end compile*)
<<CompiledFunctionTools`;
(*CompilePrint[trueCorr]*)
(*yIn=N[{3,5,7,3,8,4,6,7}];Print["yIn=",yIn];
xIn=N[{1,5,3,2,4,6,7,5}];Print["xIn=",xIn];
Print["trueCorr=",trueCorr[xIn,yIn,xIn,yIn]]//AbsoluteTimin
g
g0=trueCorr[r,r,wr,wr]/trueCorr[ones,ones,wr,wr]//AbsoluteT
iming;
Print[ListPlot[g0]];
Print[ListPlot[grr]];*)

(*****
*****
(*  define "appendCol"
*)
(*  does NOT expand table if col too long or pad col if too
short  *)
(*****
*****

```

```

appendCol[a_,b_]:=(*Module[{aa},If[Length[Dimensions[x]]<1,
aa={a};Transpose[Append[aa,b]],
  Transpose[Append[Transpose[a],b]]];*)
  If[Length[a]!=Length[b],Print["Unequal lengths. Unable to
append"];a,
  If[Length[Dimensions[a]]==1,
    Join[Transpose[{a}], Transpose[{b}],2],
    Join[a, Transpose[{b}],2]]];

(*****
*****
*)
(* define "addParam"
*)
(* if new var runs below existing cols, table is padded
line of blanks*)
(*****
*****
)
ClearAll[addParam];
addParam[a_,nameCol_,valCol_,name_,val_] :=
  Module[{nParams,ap},
    nParams=0;
    Do[If[a[[i,nameCol]]!=
"",nParams=nParams+1],{i,2,Length[a]}];

If[Length[a]==nParams+1,ap=Append[a,Table["",{i,Length[a][[1]
]}}]];ap=a];
ap=ReplacePart[ap,{nParams+2,nameCol}-> name];
ap=ReplacePart[ap,{nParams+2,valCol}-> val];
Return[ap]
];

(*****
*****
*)
(* define "mlpNeg" =-Log prob of getting negative v or h
*)
(*****
*****
)
ClearAll[mlpNeg];
mlpNeg[b_,sb_,v_]:=Log[2]-Log[Erfc[(b-v)/( $\sqrt{2}$  sb)]];

(*****
*****
*)
(* define extrapolate xc or ac curve to time zero
*)
(*****
*****
)
ClearAll[extrap];extrap[x_]:=x[[1]],x[[1]]-x[[2]],x[[1]]-

```

```

2x[[2]]+x[[3]]};

(*****
*****)
(* Fit to single exponential decay
*)
(* mode 2 □ 2p fit k-k pts & print
*)
(* mode 3 □ 3p fit k-k pts & print
*)
(* mode 20 □ 2p fit 3 pts & NO print
*)
(* mode 30 □ 3p fit 3 pts & NO print
*)
(*****
*****)
ClearAll[fitExpDecay];
fitExpDecay[iMode_, fitData_,
fitDataName_, wtData_, nEff_, kFirst_, kLast_, fitPlotName_,
isPlot_] :=
  (* mode 2 = nEff 2p; mode 3= nEff 3p *)
  Module[

{gZeroTemp, gTdTemp, gInfTemp, n, gTable, (*wTable, wTableNorm*),
iPass, eq, vars, t},

  (*Print["fit start; iMode=", iMode];*)
  saveIPrint[];
  iPrint=0;

  gTable=Table[{(i-1), fitData[[i]]}, {i, nEff}];
  wTable=wtData;
  Do[If[k<kFirst||k>kLast, wTable[[k]]=0], {k, 2, nEff}];
  n=(kLast-kFirst+1); (* num pts used HERE nEff = total
pts *)
  wTableNorm=n*wTable/Total[wTable];(*Mathematica wts must
be norm'd*)
  If[iMode==3|| iMode==30, (*s*) (* 3 PARAMS *)
  eq={gInfTemp+(gZeroTemp-gInfTemp)*Exp[-
t/gTdTemp], gTdTemp>=tauMin};
  vars={gTdTemp, gZeroTemp, gInfTemp},
  If[iMode==2||iMode==20, (* 2 PARAMS *)
  eq={gZeroTemp*Exp[-t/gTdTemp], gTdTemp>=tauMin};
  vars={gTdTemp, gZeroTemp},
  gPrint["unrecognized iMode"]; Abort[]
  ]];
  (*Print["iMode=", iMode];

```

```

Print["eq=",eq];
Print["vars=",vars];*)

iPass=0;
Monitor[
  nlm=NonlinearModelFit[
    gTable, (*data as x,y pairs*)
    eq, (* eq to use *)
    vars, (* adjustable params*)
    t, (* var in eq*)
    Weights->wTableNorm,
    StepMonitor :>{iPass=iPass+1},
    VarianceEstimatorFunction->(Total[#12 #2]/Total[#2
n/(n-2)&),
    MaxIterations->1000, (*Infinity*)
    WorkingPrecision->Automatic (*MachinePrecision*)
  ],
  Pause[0.0];
  "iPass="<>ToString[gF[iPass]]<>"
chiSq="(*<>ToString[gF[redChiSq]]*)<>"
tDecay="<>ToString[gF[gTd]]<>"
gInf="<>ToString[gF[g0]]<>ToString[gF[gInf]]<>" gAmp="
  ];
  (* gPrint["expModelDecay: end NLM"];*)

  gPrint["\n","fit results"];
  gPrint["paramTable=",paramTable=nlm["ParameterTable"]];
  (*val,SE,p*)
  gPrint["params=",params=nlm["BestFitParameters"]];
  gPrint["errs=",errs=nlm["ParameterErrors"]];

gPrint["estdSD=",estdSD=\[Sqrt]nlm["EstimatedVariance"]];
gPrint["nlm[0]=",nlm[0]]; (*"FittedModel"*)

gSD=estdSD;
gTd=params[[1,2]];
gTdErr=errs[[1]];
gZero=params[[2,2]];
gZeroErr=errs[[2]](*errs[[2]]*);
If[(iMode==2)||iMode==20, (*2p fits*)
  gInf=0;(*params[[2,2]];*)
  gInfErr=0,
  If[iMode==3 ||iMode==30,(*s*)
    gInf=params[[3,2]]; (*3p fit*)
    gInfErr=errs [[3]],
    Print["unrecognized iMode. Exiting...."];Abort[]
  ]];

```

```

dP["\n",gTd];
dP[gTdErr];
dP[gZero];
dP[gZeroErr];
dP[gInf];
dP[gInfErr];

If[isPlot&&(iMode==2||iMode==3),
  fluctCalc=Table[nlm[i-1],{i,1,nEff}];
  xyCalc=Table[{i-1,nlm[i-
1]},{i,kLast}];range={1.1Min[xyCalc[[All,2]],gTable[[All,2]
]],1.1*Max[xyCalc[[All,2]],gTable[[All,2]]]};

fitPlot=ListPlot[{gTable[[kFirst;;kLast,2]],xyCalc[[kFirst;
;kLast,2]]},PlotLabel-> fitPlotName]
];(*if*)
restIPrint[];
(*Pause[1*^9];*)
(*Print["fit exit"];*)
];

(*setIPrint[0];fitExpDecay[fitData,
fitDataName,wtData,nEff,kFirst,kLast,fitPlotName];restIPrint
[];
Pause[1*^9];*)

(*****
*****)
(* do exp fit 3 Pts *)
(*****
*****)
ClearAll[fitAnisDecay3Pts];
fitAnisDecay3Pts[:=(
(* iMode 2= k-k pts 2p; iMode 3= k-k Pts 3p *)
(* iMode=20 > 2pts; iMode=30 > 3 pts NO PRINT *)
iMode=10*nParams;
wTable=gwrwr;

fitExpDecay[iMode,grrFluct12,"grrFluct12",wTable,nEff,2,4,N
ull,False];
grZero3Pts=gZero;
grZeroErr3Pts=gZeroErr;
grTd3Pts=gTd;
grTdErr3Pts=gTdErr;
grInf3Pts=gInf;
grInfErr3Pts=gInfErr;
grSD3Pts=gSD);

```

```

(*****
*****)
(* print anis fit nums
*)
(*****
*****)
ClearAll[printAnisFitNums];
printAnisFitNums[]:=Module[{},
Print[
"tau=",gF[grTdFull], " +/- ",gF[grTdErrFull],
" g0=",gF[grZeroFull], " +/- ",gF[ grZeroErrFull],
" g∞=", gF[grInfFull]," +/- ", gF[grInfErrFull],
" gSD=",gF[grSDFull], " nEff=",gF[ nEff]]];

(*****
*****)
(*getFileList[]: get names of raw data files in current
directory *)
(*****
*****)
ClearAll[getFileList];
getFileList[]:=Module[{f,l,c1,c2,c3,c4,c,i},
f=FileNames[All];l=Length[f];
c1=StringContainsQ[f, "LRB"];c2=StringContainsQ[f,
"csv"];
c3=Thread[!StringContainsQ[f,
"results"]];c4=Thread[!StringContainsQ[f, "plots"]];
c=Table[c1[[i]]&& c2[[i]]&& c3[[i]]&& c4[[i]],{i,l}];
fs={};For [i=1,i<=l,i++,If[c[[i]],AppendTo[fs,f[[i]]]];
Return[fs]
];
(*getFileList[]*)

(*****
*****)
(* wtd linear fit
*)
(*****
*****)
ClearAll[probs,xList,data,wtdLin];
wtdLinFit[s_,d_,nP_,iFirstOrder_,wTable_]:=Module[{x,xList,
data,ev, β,se,pr,probs,lm},
(* nP=num params; iFirstOrder=*)
n=Length[s];
data=Transpose@{s,d};
(*Print[ListPlot[data]];*)

```

```

xList=Table[xi, {i, iFirstOrder, nP+iFirstOrder-1}];
(*Print[xList];*)
lm=LinearModelFit [data, xList, x,
  IncludeConstantBasis->False,
  Weights->wTable,
  VarianceEstimatorFunction->((Total[#12
#2]/Total[#2])*(n/(n-2))&)];
β=lm["BestFitParameters"];
ev=lm["EstimatedVariance"];
se=lm["ParameterErrors"]; (* std err*)
pr=lm["PredictedResponse"]; (* yCalc*)
probs= (1/2)(β/se)2;
(*dP[probs];
dP[Total[probs]];*)
Return[{ev, β, se, (1/2)(β/se)2}]
(* [[1]]= fit variance, [[2]]=params, [[3]]=std errs,
[[4]]=mlp param probs *)
];

(*****
**)
(* functiot to calc variance of variance of list
*)
(*****
**)
ClearAll[varOfVar];
varOfVar[x_]:=Module[{n,
μ, ones, dxSq, μ2, μ4, μ2Sq, varVar, vov},
  n=Length[x];
  μ=Total[x]/n;
  ones=Table[1, {n}];
  dxSq=(x-ones*μ)2;
  μ2=Total[dxSq]/n;
  μ4=Total[dxSq2]/n;
  μ2Sq=μ2;
  vov=(μ4-μ2Sq)/n+μ2Sq*(2/(n(n-1)))
];

(* (*****
****)
(* test var of difference in variances
*)
(*****
**)
ClearAll[x, y, vx, vy, diff, vvx, vvy, vvsum, vvsumsd, ratio];
n=1000;
m=1000;

```

```

iPrint=0;
x=RandomVariate[NormalDistribution[10,1],n];
y=RandomVariate[NormalDistribution[10,1],n];
xt=Table[x=RandomVariate[NormalDistribution[10,1],n],
m]; (*dP[xt];*)
yt=Table[y=RandomVariate[NormalDistribution[10,1],n],
m]; (*dP[yt];*)
s2x=Variance[xt[]];dP[s2x];
s2y=Variance[yt[]];dP[s2y];
vovx=Table[varOfVar[xt[[i]]],{i,m}];dP[vovx];
vovy=Table[varOfVar[yt[[i]]],{i,m}];dP[vovy];
diff=s2x- s2y; dP[diff];
vvsum=vovx+vovy;dP[vvsum];
vvsumsd=Sqrt[vvsum];dP[vvsumsd];
ratio=(diff/vvsumsd);dP[ratio];
mlp=ratio^2/2;dP[mlp];
iPrint=1;
Length[ratio]
Variance[ratio]
Histogram[ratio]*)

restIPrint[]; (* overall for 'definitions'*)

(*****
**)
(*           end of definitions
*)
(*****
**)

ClearAll[depend];
depend[vEff_List,hEff_List,bvEff_List,bhEff_List,iWt_Integer,
iGeom_Integer,kFirst_Integer,kLast_Integer,nBkgsDummy_Integer,
gDummy_Real,bDummy_Real,cDummy_Real,fDummy_Real,iFirstDot_Integer,
iLastDot_Integer]:=Module[({*nMax,n,m,nEff,nCorr,kMax,kMin,nk,nBkgs*}),

(*****
*****
*) (* initialization code
*) (*****
*****
saveIPrint[];
(*dP[isNewMin," ",iPass]; *)
If [Length[vEff]!= Length[hEff], Print["vEff and hEff
unequal length. Exiting..."];Abort[] ];
nEff=Length[vEff];

```

```

zeros=Table[0,{nEff}];
ones=Table[1,{nEff}];
kList=Table[nEff-k+1,{k,1,nEff}];
sqrtKList= $\sqrt{kList}$  ;
time=Table[i, {i,nEff}];
kMax=If[kLast==0, nEff, kLast];
If [kLast>nEff,Print["kMax exceeds nEff.
Exiting..."];Abort[]];
kMin=If[kFirst==0, 2,kFirst];
nk=kMax-kMin+1;
(*nBkgs=If [nBkgsDummy<0,4,nBkgsDummy];*)

(*****
*****
  (* set constants
*)

(*****
*****
  g=gDummy; (* old g-fact: (1-g)v, (1+g)h *)
  b=bDummy; (* sample bkg *)
  c=cDummy; (* camera bkg *)
  f=fDummy; (* ???? *)
  (*Pause[1*^9];*)

(*****
*****
  (* CALCULATION STARTS HERE
*)

(*****
*****

(*****
****)
  (* get inten funct & baseline vars
*)

(*****
****)
  (* nBkgs = number of baselines averaged *)
  bvAvg=Mean[bvEff]; (* avg of nBkgs traces *)
  bhAvg=Mean[bhEff];

```

```

If[ f<0,f=cBkg];(*if f<0 (e.g. -1), f initialized to
cBkg *)
If [isBkgs!=0, (*yes, bkgs. calc inten funct from bkgs
*)
  intV=(bvEff-f)/(bvAvg-f);(*inten should=iv,ih*)
  intH=(bhEff-f)/(bhAvg-f),(*inten should=iv,ih*)
  (*else let inten factors=1 *)
  intV=Table[1,{nEff}];
  intH=intV
  ];
(*dP[intH];
Pause[1*^9];*)

bvc=(1-g)bvEff/intV; (* correcting for src fluct * )
bhc=(1+g)bhEff/intH; (* no corr if no bKgs *)

If [isBkgs!=0,(*yes, bkgs. calc bkg variances
UNCORRECTED *)
  e2bv=Variance[bvEff]; (* actual variance corr'd bkgs *)
  e2bh=Variance[bhEff],
  (*else*)
  e2bv=1; (* need some placeholder num here *)
  e2bh=1
  ];

If [isBkgs!=0,(*yes, bkgs. calc CORRECTED bkg variances
*)
  e2bvc=(1-g)^2 Variance[bvEff]; (* actual variance corr'd
bkgs *)
  e2bhc=(1+g)^2 Variance[bhEff],
  (*else*)
  e2bvc=1; (* need some placeholder num here *)
  e2bhc=1
  ];

If [iWt==2, (* rs weights*)
  (*Gaussian*)vVar=e2bvc;hVar=e2bhc,
  If[iWt==1,
  (*Poisson*)vVar=(1-g)^2 (vEff+2 bvEff); (* 2 baselines /
vc*)hVar=(1+g)^2 (hEff+2 bhEff),
  Print["bad iWt"]]
  ];
vVarSq=vVar^2;
hVarSq=hVar^2;

(*intV=(bvEff-f)/(bvAvg-f);(*inten should=iv,ih*)
intH=(bhEff-f)/(bhAvg-f);*)

```

```

vs=(vEff-bvEff)/intV; (* v sub't'd *)
hs=(hEff-bhEff)/intH;
vM=Mean[vs]; (* v(sub't'd)Mean *)
hM=Mean[hs];
vSD=StandardDeviation[vs];
hSD=StandardDeviation[hs];

g= (vSD-hSD)/(vSD+hSD);
c=(hM(1+g)-(1-g) vM)/(2g);
(*Print[iPass," ",vM," ",hM," ",vSD," ",hSD," ",g,"
",c," ",b];*)
vc=Collect[Simplify[(1-g)(vs-c)-b/2],vs];
hc=Collect[Simplify[(1+g)(hs-c)+b/2],hs];

s=vc+iGeom*hc;sSq=s2;s4th=sSq2;
d=vc-hc;dSq=d2;d4th=dSq2;
r=Table[If[s[[i]]!=0,d[[i]]/s[[i]],0], {i,nEff}];rSq=r2;

(*****
****)
(* get variances and wts
*)

(*****
****)
If[(isNewMin!=0)|| (iPass==0 ),(*dP[iPass];*)
(*Print[iPass];*)
sVar=ones*(2(vVar+iGeom2 hVar)); (* two bkgs each
meas't*)
dVar=ones*(2(vVar+hVar))
];

(*get INITIAL values for sAvg, rAvg to calc init rWts*)
If[iPass== 0,
ws=ones/sVar;
sws=s*ws;
sAvg=Total[sws]/Total[ws];
rAvg=zeros (* start with rAvg = 0*)
];

wsRaw=ones/sVar;
ws=nEff*wsRaw/Total[wsRaw]; (*normalize s-wts*)
sws=s*ws;
sSqws=sSq*ws;
sAvg=Total[sws]/Total[ws];

```

```

wdRaw=ones/dVar;
wd=nEff*wdRaw/Total[wdRaw]; (*normalize d-wts*)
dwd=d*wd;
dSqwd=dSq*wd;
dAvg=Total[dwd]/Total[wd];

(* only have 2nd order sums of v, h so ignore 4th-order
tems *)
z=iGeom;
rVar=If[iWeight==1,ones,If
[iWeight==2,1/sSq,If[iWeight==3,
(1/s4th)*(vVar (-1+r)^2 sSq+hVar sSq (1+r z)^2 +3
vVarSq (-1+r)^2+3 z^2 hVarSq (1+r z)^2 +3 vVar hVar ((1-4
z+z^2 )+6 r z (1- z) +6 z^2 rSq )],Print["Bad iWeight"]]]];
wrRaw=1/rVar;
wr=nEff*wrRaw/Total[wrRaw]; (*normalize r-wts*)
rwr=r*wr;
rSqwr=rSq*wr;
rAvg=Total[rwr]/Total[wr];

(*Print["end calc basic terms"];*)

(*****
***)
(* calc correlations (now AVERAGES )
*)

(*****
***)

(*kList=Table[nEff-k+1,{k,1,nEff}];*)
(*oneList=Table[1,{k,1,nEff}];*)
ClearAll[];
sws=s*ws;
rwr=r*wr;
(*sAvg=Total[sws]/Total[ws];
rAvg=Total[rwr]/Total[wr];*)
(*g11=ListCorrelate[ones,ones,1,0];*)

gss=ListCorrelate[sws,sws,1,0];
gsr=ListCorrelate[sws,rwr,1,0];
grs=ListCorrelate[rwr,sws,1,0];
grr=ListCorrelate[rwr,rwr,1,0];

gs1=ListCorrelate[sws,ones,1,0];

```

```

gls=ListCorrelate[ones,sws,1,0];
grl=ListCorrelate[rwr,ones,1,0];
glr=ListCorrelate[ones,rwr,1,0];

gsws=ListCorrelate[ws,ws,1,0];
gswr=ListCorrelate[ws,wr,1,0];
gwrws=ListCorrelate[wr,ws,1,0];
gwrwr=ListCorrelate[wr,wr,1,0];

gws1=ListCorrelate[ws,ones,1,0];
glws=ListCorrelate[ones,ws,1,0];
gwrl=ListCorrelate[wr,ones,1,0];
glwr=ListCorrelate[ones,wr,1,0];
g11=ListCorrelate[ones,ones,1,0];

s1Avg=gs1/gws1;
s2Avg=gls/glws;
r1Avg=gr1/gwrl;
r2Avg=g1r/glwr;

gssFluct12=(g11 *gss-gs1*gls)/(gws1 glws);
gsrFluct12=(g11 *gsr-gs1*g1r)/(gws1 glwr);
grsFluct12=(g11 *grs-gr1*gls)/(gwrl glws);
grrFluct12=(g11 *grr-gr1*g1r)/(gwrl glwr);
(*
gssFluct12=gss/gsws-s1Avg*s2Avg;
gsrFluct12=gsr/gswr-s1Avg*r2Avg;
grsFluct12=grs/gwrws-r1Avg*s2Avg;
grrFluct12=grr/gwrwr-r1Avg*r2Avg;*)

(* (***** enforce that vVar = hVar
*****)
x=StandardDeviation [vs];
y=StandardDeviation[hs];
g=(x-y)/(x+y);*)

(***** prob the neg pts are really zero
*****)
(* assume true val of -'ve pt actually zero *)
mlpNegVSum=Total[0.5*(1-Sign[vc])*vc^2/(2*e2bvc)];
mlpNegHSum=Total[0.5*(1-Sign[hc])*hc^2/(2*e2bhc)];
mlpNeg=mlpNegVSum+ mlpNegHSum; (* vc,hc must both be
non-neg *)
(*Print[vc[[10]]," ",mlpNegVSum," ",mlpNegHSum,"
",mlpNeg, " iWt=",iWt];*)

(***** prob that s,r uncorrelated

```

```

*****
(* all "mlp" terms preceded by (1/2)Log[2 π]-
(1/2)Log[nEff] *)
(* distCorr=distCorrFunct[r,s];*)

(*****
****)
RSDistCorr=distCorrFunct[r,s];
RSPmCorr= wtdCorrCoeff[r,s,ones];
RSWpmCorr= wtdCorrCoeff[r,s,wr*ws];
mlpRSDistCorr=(1/2)RSDistCorr/(1(*-rho^2*)) nEff;
mlpRSPmCorr=(1/2)RSPmCorr^2 / (1(*-rho^2*)) nEff;
mlpRSWpmCorr=(1/2)RSWpmCorr^2/( 1(*-rho^2*)) nEff;

(* (***** prob coeffs a0=0,a1=0 ***** )
wTable=ws*wr;
mlpRAvg=wtdLinFit[s,r,1,0,wTable][[4]][[1]];
mlpRSlope=wtdLinFit[s,r,1,1,wTable][[4]][[1]];
(*mlpRCurve=wtdLinFit[s,r,1,2,wTable][[4]][[1]];*)
(*mlpR4th=wtdLinFit[s,r,1,3,wTable][[4]][[1]];*)*)
wTable=ws*wr;
mlpRAvg=0;
mlpRSlope=wtdLinFit[s,r,1,1,wTable][[4]][[1]];
mlpRCurve=0;

(* (***** enfoprob that v- & h-variances are same
*****)
varDiff=Variance[vc]-Variance[hc];
varDiffErr=√((varOfVar[vc]+varOfVar[hc])/1(*only 1
pt*));
mlpVarDiff=(1/2)(varDiff/varDiffErr)^2; (* d=(vc-bvc)-
(hc-bhc) OK!!!! *)
(*Print["dAvg=", dAvg," dAvgErr=",dAvgErr,"
mlpDAvg1P=",mlpDAvg1P];*)*)
mlpVarDiff=0;

(* (***** prob that true min is zero ***** )
nMin=16;
vcs=Sort[vc];
hcs=Sort[hc];
mlpZeroErr=Sum[vcs[[i]]^2/(2*varV)+
hcs[[i]]^2/(2*varH),{i,1,nMin}];*)
mlpZeroErr=0;

(***** overall probability ***** )
(* all selected measures *)
mlpTot=0;

```

```

    mlpTot=mlpTot+0mlpNeg;          (*prob neg pts are really
zero*)
    mlpTot=mlpTot+0mlpRSDistCorr;   (*prob that r,s
uncorrelated*)
    mlpTot=mlpTot+0mlpRSPmCorr;     (*prob that r,s
uncorrelated; wtd*)
    mlpTot=mlpTot+1mlpRSWpmCorr;    (*prob that r,s
uncorrelated; wtd*)

    mlpTot=mlpTot+0 mlpRAvg;        (*prob that intercept is
zero*)
    mlpTot=mlpTot+0mlpRSlope;       (*prob that slope is zero*)
    mlpTot=mlpTot+0mlpRCurve;       (*prob that slope is zero*)
    (*mlpTot=mlpTot+1mlpRCurve;*)   (*prob that slope is
zero*)

    mlpRCurve=mlpZeroErr;
    mlpTot=mlpTot+0 mlpVarDiff;
    mlpTot=mlpTot+0 mlpZeroErr;

    mlpTot=mlpTot(*+0 p3*);         (*max of 3 adj constants*)

    Return[mlpTot]
]; (*module*)
(*inFileName="Ige-1-LRB.csv";
data=Import[inFileName];
bvEff=data[[2;;All,2]];
bhEff=data[[2;;All,3]];
vEff=data[[2;;All,8]]; hEff=data[[2;;All,9]];
dep=depend[vEff,hEff,bvEff,bhEff,2,1, 0,0,4,+0.4,0.,0.,-
1.,iFirstDot,iLastDot];
Print["nEff=",nEff, " dep=", dep];
isNewMin=1;
depend[];*)

restIPrint[];

(*****
***)
(* end of "depend" code
*)
(*****
***)

(*****
*****
(* maximize prob wrt g,b,c

```

```

*)
(*****
*****)
ClearAll[optimize ];
optimize[]:=Module[{(*iWt=2,
iFirst=1,iLast=1000,kFirst=0,nBkgs=4,kLast=1000,iThreshMode
=0,thresh=0,iOrderSave=4,iGeom=1,g0=.40,b0=0.,c0=0.,f0=-
1.*)},

    saveIPrint[];
    (*progressFileName=base<>"-"<>"progress"<>"-
"<>DateString[]<>".csv";
    If [FileExistsQ[progressFileName],
    Close[progressFileName];
    DeleteFile[progressFileName]];
    prog=OpenWrite [progressFileName]; *)

pInit=depend[vEff,hEff,bvEff,bhEff,iWt,iGeom,0,0,nBkgs,g0,b
0,c0,f0,iFirstDot,iLastDot];
    (*Print["pInit=",pInit," g0=",g0," b0=", b0,"
c0=",c0," f0=", f0," iWt=",iWt];*)

    pMin=pInit;
    iPass=0;
    qq={0,{{0,0},{0,0},{0,0},{0,0}}};
    Off[StringJoin::string];
    c=c0;
    f=f0;
    isNewMin=1;
    qq=Monitor[

FindMinimum[{depend[vEff,hEff,bvEff,bhEff,iWt,iGeom,0,0,nBk
gs,g,b,c,f,iFirstDot,iLastDot],(*b≥ 0&&c≥ 0&&*) -1<g<1(*&&
-1<f<1*)},{(*{g,g0}*)(*,*){b,b0}{*,{c,c0}*)(*,{f,f0}*)},
    EvaluationMonitor:>
    (
    p=mlpTot;
    iPass=iPass+1;
    (*WriteString[prog,
gF[iPass],",",gF[p],",",gF[mlpNeg],",",gF[mlpBkgs],",",gF[m
lpRAvg],",",gF[mlpDAvg],",",gF[mlpRhoSR],",",gF[mlpRhoRS],",
",gF[mlpRhoSSR],",",gF[mlpRhoSRR],",",gF[g],",",gF[c],",",g
F[b],"\n"];*)
    If[p<pMin ,
    gMin=g;bMin=b;cMin=c;pMin=p;fMin=f;
    isNewMin=1(*;dP[iPass]*),

```

```

        (*else*) isNewMin=0;];
    If[p<pGoal|| iPass>= maxPass, Goto[continue]];
    If
[CurrentValue["ControlKey"]==True,Goto[continue]]
    ), (* end EvalMon*)
    PrecisionGoal->Automatic (* tried 0 *),
    AccuracyGoal ->Automatic (* tried 3 *),
    MaxIterations->1000(*Infinity*),
    WorkingPrecision->Automatic (*"MachinePrecision"*),
    Method->Automatic
    ], (* end FindMin*)
    "iPass="<>ToString[gF[iPass]]
    <>" iDot=" <>ToString[gF[iDot]]
    <>" of " <>ToString[gF[nDots]]

    <>"\n RAvg="<> ToString[gF[mlpRAvg]]
    <>" RSlope="<> ToString[gF[mlpRSlope]]
    <>" RCurve="<> ToString[gF[mlpRCurve]]
    <>" varDiff="<> ToString[gF[mlpVarDiff]]
    <>" zeroErr="<> ToString[gF[mlpZeroErr]]

    <>"\n mlpNeg=" <> ToString[gF[mlpNeg]]
    <>" RSDistCorr=" <> ToString[gF[mlpRSDistCorr]]
    <>" RSPmCorr=" <> ToString[gF[mlpRSPmCorr]]
    <>" RSWPmCorr=" <> ToString[gF[mlpRSWpmCorr]]

    <>"\n p="<>ToString[gF[p]]
    <>" f="<>ToString[gF[f]]
    <>" g="<>ToString[gF[g]]
    <>" c="<>ToString[gF[c]]
    <>" b="<>ToString[gF[b]]

]; (*end Monitor*)

Label[continue ];
(*Print["\n iPass=",iPass," pMin=",pMin," gMin=",gMin,"
bMin=",bMin," cMin=", cMin, " fMin=",fMin];*)
(*Print["pause"];
Pause[1*^6];*)

If[qq[[1]]==0&& p>pGoal,
    Print["Optimization NOT completed! " ],
    (*otherwise*)
    Print["Optimization successful! - p = ",p]
];

p=depend[vEff,hEff,bvEff,bhEff,iWt,iGeom,kFirst,kLast,nBkgs

```



```

cNew=cMin;
fNew=fMin;

(*depend[vEff_List,hEff_List,bvEff_List,bhEff_List,iWt_Integer,iGeom_Integer,kFirst_Integer,kLast_Integer,nBkgsDummy_Integer,gDummy_Real,bDummy_Real,cDummy_Real,fDummy_Real,iFirstDot_Integer,iLastDot_Integer]*)
(Print["pOpt=",depend[vEff,hEff,bvEff,bhEff,iWt,iGeom,kFirst,kLast,nBkgs,gNew,bNew,cNew,fNew,iFirst,iLastDot],
" gNew=",gNew," bNew=", bNew," cNew=",cNew," fNew=",fNew]*);
SetOptions[{Plot,ListPlot},ImageSize->{144,108}];
ListPlot[{vc,hc},PlotRange->Automatic(*{0,100}*),PlotLabel->"vc,hc"];

gsrPlots={gsrFullPlot=ListPlot[{gsrFluct12[[2;;]]},PlotLabel->"gsrFluctFull"],
gsr500Plot=ListPlot[{gsrFluct12[[2;;500]]},PlotLabel->"gsrFluct500"],
gsr200Plot=ListPlot[{gsrFluct12[[2;;200]]},PlotLabel->"gsrFluct200"],
gsr100Plot=ListPlot[{gsrFluct12[[2;;100]]},PlotLabel->"gsrFluct100"]};
gPrint[gsrPlots];

grsPlots={grsFullPlot=ListPlot[{grsFluct12[[2;;]]},PlotLabel->"grsFluctFull"],
grs500Plot=ListPlot[{grsFluct12[[2;;500]]},PlotLabel->"grsFluct500"],
grs200Plot=ListPlot[{grsFluct12[[2;;200]]},PlotLabel->"grsFluct200"],
grs100Plot=ListPlot[{grsFluct12[[2;;100]]},PlotLabel->"grsFluct100"]};
gPrint[grsPlots];

gssPlots={gssFullPlot=ListPlot[{gssFluct12[[2;;]]},PlotLabel->"gssFluctFull"],
gss500Plot=ListPlot[{gssFluct12[[2;;500]]},PlotLabel->"gssFluct500"],
gss200Plot=ListPlot[{gssFluct12[[2;;200]]},PlotLabel->"gssFluct200"],
gss100Plot=ListPlot[{gssFluct12[[2;;100]]},PlotLabel->"gssFluct100"]};
gPrint[gssPlots];

```

```

grrPlots={grrFullPlot=ListPlot[{grrFluct12[[2;;]]},PlotLabel-
l->"grrFluctFull"],
  grr500Plot=ListPlot[{grrFluct12[[2;;500]]},PlotLabel-
>"grrFluct500"],
  grr200Plot=ListPlot[{grrFluct12[[2;;200]]},PlotLabel-
>"grrFluct200"],
  grr100Plot=ListPlot[{grrFluct12[[2;;100]]},PlotLabel-
>"grrFluct100"]};
gPrint[grrPlots];

  restIPrint[];
  );
(*showOptResults[]*)

(*****
*****
(*ANIS: fit anis decays of various ranges
*)
(*****
*****
ClearAll[fitAnisDecay];
fitAnisDecay[]:= (
  saveIPrint[];
  iPrint=0;
  grFitData=grrFluct12;
  grFitDataName="grrFluct";
  wrr=gwrwr;
  ClearAll[grFitPlots,grFitPlotFull,grFitPlot500,
grFitPlot200, grFitPlot100];
  If[nParams==2,iMode=2,If[nParams==3,iMode=3,Print
["fitAnisDecay: bad nParams"]]];

  grFitFull=fitExpDecay[iMode,grFitData,
grFitDataName,wrr,nEff,2,Length[grFitData],"grFitFull",True
];
  grkFirstFull=2;grkLastFull=Length[grFitData];

grSDFull=gSD;grInfFull=gInf;grZeroFull=gZero;grTdFull=gTd;

grInfErrFull=gInfErr;grZeroErrFull=gZeroErr;grTdErrFull=gTd
Err;

grFluctCalcFull=fluctCalc;grFitPlotFull=fitPlot;(*dP[gTdErr
]; dP[grTdErrFull];*)

```

```

    grFit500=fitExpDecay[iMode,grFitData,
grFitDataName,wrr,nEff,2,500,"grFit500",True];
    grkFirst500=2;grkLast500=500;
    grSD500=gSD;grInf500=gInf;grZero500=gZero;grTd500=gTd;

grInfErr500=gInfErr;grZeroErr500=gZeroErr;grTdErr500=gTdErr
;
    grFluctCalc500=fluctCalc;grFitPlot500=fitPlot;(*
dP[gTdErr]; dP[grTdErr500];*)

    grFit200=fitExpDecay[iMode,grFitData,
grFitDataName,wrr,nEff,2,200,"grFit200",True];
    grkFirst200=2;grkLast200=200;
    grSD200=gSD;grInf200=gInf;grZero200=gZero;grTd200=gTd;

grInfErr200=gInfErr;grZeroErr200=gZeroErr;grTdErr200=gTdErr
;
    grFluctCalc200=fluctCalc;grFitPlot200=fitPlot;

    grFit100=fitExpDecay[iMode,grFitData,
grFitDataName,wrr,nEff,2,100,"grFit100",True];
    grkFirst100=2;grkLast100=100;
    grSD100=gSD;grInf100=gInf;grZero100=gZero;grTd100=gTd;

grInfErr100=gInfErr;grZeroErr100=gZeroErr;grTdErr100=gTdErr
;
    grFluctCalc100=fluctCalc;grFitPlot100=fitPlot;

    Print[grFitPlots={grFitPlotFull,grFitPlot500,
grFitPlot200, grFitPlot100 }];
    restIPrint[]
    );
(*fitAnisDecay[]*)

(*****
*****
(*INTEN:  fit single exponential decay
*)
(*****
*****

ClearAll[fitIntenDecay];
fitIntenDecay[:=(
    saveIPrint[];
    iPrint=0;
    gsFitData=gssFluct12;
    gsFitDataName="gssFluct";

```

```

wss=gwsws;

If[nParams==2,iMode=2,If[nParams==3,iMode=3,Print ["s
fitexp decay - bad nParams"]]];
gsFitFull=fitExpDecay[iMode,gsFitData,
gsFitDataName,wss,nEff,2,nEff,"gsFitFull",True];
gskFirstFull=2;gskLastFull=999;

gsSDFull=gSD;gsInfFull=gInf;gsZeroFull=gZero;gsTdFull=gTd;

gsInfErrFull=gInfErr;gsZeroErrFull=gZeroErr;gsTdErrFull=gTd
Err;
gsFluctCalcFull=fluctCalc;gsFitPlotFull=fitPlot;

gsFit500=fitExpDecay[iMode,gsFitData,
gsFitDataName,wss,nEff,2,500,"gsFit500",True];
gskFirst500=2;gskLast500=500;
gsSD500=gSD;gsInf500=gInf;gsZero500=gZero;gsTd500=gTd;

gsInfErr500=gInfErr;gsZeroErr500=gZeroErr;gsTdErr500=gTdErr
;
gsFluctCalc500=fluctCalc;gsFitPlot500=fitPlot;

gsFit200=fitExpDecay[iMode,gsFitData,
gsFitDataName,wss,nEff,2,200,"gsFit200",True];
gskFirst200=2;gskLast200=200;
gsSD200=gSD;gsInf200=gInf;gsZero200=gZero;gsTd200=gTd;

gsInfErr200=gInfErr;gsZeroErr200=gZeroErr;gsTdErr200=gTdErr
;
gsFluctCalc200=fluctCalc;gsFitPlot200=fitPlot;

gsFit100=fitExpDecay[iMode,gsFitData,
gsFitDataName,wss,nEff,2,100,"gsFit100",True];
gskFirst100=2;gskLast100=100;
gsSD100=gSD;gsInf100=gInf;gsZero100=gZero;gsTd100=gTd;

gsInfErr100=gInfErr;gsZeroErr100=gZeroErr;gsTdErr100=gTdErr
;
gsFluctCalc100=fluctCalc;(gsFitPlot100=fitPlot);

gPrint[gsFitPlots={gsFitPlotFull,gsFitPlot500,
gsFitPlot200,gsFitPlot100}];
restIPrint[];
);

(*setiPrint[0];fitIntenDecay[];restIPrint[];*)

```

```

(*****
*****)
(* save results as CSV
*)
(*****
*****)

ClearAll[saveResults];
saveResults[:=(
  (* generate timestamp for all output this pass *)
  saveIPrint[0];
  isStamp=True;
  base=StringTake[inFileName, StringLength[inFileName]-
4]<>If[iDot!=-1, "-
d"<>IntegerString[iDot,10,2],""];tdList=DateList[];
  dateFormat={"Year", "-", "Month", "-", "Day"};
  timeFormat={"Hour", ":", "Minute", ":", "Second"};
  dateString=DateString[tdList, dateFormat];
  timeString=DateString[tdList, timeFormat];
  timeStampFormat={"Year", (*"-", *) "Year", (*"-", *) "Month", (*"-
", *) "Day", "-", "Hour24", "Minute", "Second"};

timeStampString=If[isStamp, DateString[tdList, timeStampForma
t], ""];
  (* dP[timeStampString];*)

  (* set up results file *)
  resultsFileName=base<>"-"<>"results"<>"-
"<>timeStampString<>".csv";

  If [FileExistsQ[resultsFileName],
    q=Close[resultsFileName];
    gPrint[q];
    DeleteFile[resultsFileName]
  ];
  res=OpenWrite[resultsFileName];
  nameCol=1;
  valCol=2;

ClearAll[rA]; (* rA= "Results Array" *)
(* fill columns C through V with all pts *)
rA=Prepend[Table["", {i, nEff}], "param"];
rA=appendCol[rA, Prepend[Table["", {i, nEff}], "value"]];
rA=appendCol[rA, Prepend[Table[i, {i, nEff}], "iPt"]];
rA=appendCol[rA, Prepend[vEff, "vEff"]];
rA=appendCol[rA, Prepend[hEff, "hEff"]];

```

```

rA=appendCol[rA,Prepend[bvEff, "bvEff"]] ;
rA=appendCol[rA,Prepend[bhEff, "bhEff"]] ;
rA=appendCol[rA,Prepend[vc, "vc"]] ;
rA=appendCol[rA,Prepend[hc, "hc"]] ;
rA=appendCol[rA,Prepend[s, "s"]] ;
rA=appendCol[rA,Prepend[d, "d"]] ;
rA=appendCol[rA,Prepend[r, "r"]] ;
rA=appendCol[rA,Prepend[wr, "wr"]] ;
(*rA=appendCol[rA,Prepend[gssFluct, "gssFluct"]] ;*)
rA=appendCol[rA,Prepend[gssFluct12, "gssFluct12"]];
(*rA=appendCol[rA,Prepend[gsrFluct, "gsrFluct"]] ;*)
rA=appendCol[rA,Prepend[gsrFluct12, "gsrFluct12"]];
(*rA=appendCol[rA,Prepend[grsFluct, "grsFluct"]] ;*)
rA=appendCol[rA,Prepend[grsFluct12, "grsFluct12"]];
(*rA=appendCol[rA,Prepend[grrFluct, "grrFluct"]] ;*)
rA=appendCol[rA,Prepend[grrFluct12, "grrFluct12"]];
rA=appendCol[rA,Prepend[wrr, "wrr"]] ;
rA=appendCol[rA,Prepend[grFluctCalcFull,
"grrFluctCalcFull"]] ;
rA=appendCol[rA,Prepend[grFluctCalc500,
"grrFluctCalc500"]] ;
rA=appendCol[rA,Prepend[grFluctCalc200,
"grrFluctCalc200"]] ;
rA=appendCol[rA,Prepend[grFluctCalc100,
"grrFluctCalc100"]];

(* fill cols A and B with names and values of params
*)rA=addParam[rA,nameCol,valCol,"nb",NotebookFileName[]];rA
=addParam[rA,nameCol,valCol,"inFileName",inFileName];
rA=addParam[rA,nameCol,valCol,"iDot",iDot];
rA=addParam[rA,nameCol,valCol,"nEff",nEff];
rA=addParam[rA,nameCol,valCol,"cBkg",cBkg];
rA=addParam[rA,nameCol,valCol,"thMode",0]; (* no longer
used*)
rA=addParam[rA,nameCol,valCol,"thLev",0]; (* no longer
used*)
rA=addParam[rA,nameCol,valCol,"nCorr",0]; (* no longer
used*)
rA=addParam[rA,nameCol,valCol,"n2v",0]; (* no longer
used*)
rA=addParam[rA,nameCol,valCol,"n2h",0]; (* no longer
used*)
rA=addParam[rA,nameCol,valCol,"n2vc",0]; (* no longer
used*)
rA=addParam[rA,nameCol,valCol,"n2sc",0]; (* no longer
used*)

```

```

rA=addParam[rA,nameCol,valCol," "," "];
rA=addParam[rA,nameCol,valCol,"kMin",kMin];
rA=addParam[rA,nameCol,valCol,"kMax",kMax];

rA=addParam[rA,nameCol,valCol,"mlpIntercept",mlpRAvg(*mlpIntercept*)];

rA=addParam[rA,nameCol,valCol,"mlpSlope",mlpRSlope(*mlpSlope*)];

rA=addParam[rA,nameCol,valCol,"mlpLFeedthru",mlpVarDiff];
rA=addParam[rA,nameCol,valCol,"mlpTot",mlpTot];
rA=addParam[rA,nameCol,valCol,"unused1",0];
rA=addParam[rA,nameCol,valCol,"unused2",0];
rA=addParam[rA,nameCol,valCol,"iPass",iPass];
rA=addParam[rA,nameCol,valCol,"pMin",pMin];
rA=addParam[rA,nameCol,valCol,"gMin",gMin];
rA=addParam[rA,nameCol,valCol,"bMin",bMin];
rA=addParam[rA,nameCol,valCol,"cMin",cMin];
rA=addParam[rA,nameCol,valCol,"fMin",fMin];

rA=addParam[rA,nameCol,valCol," "," "];
rA=addParam[rA,nameCol,valCol,"rFitData",grFitDataName];

rA=addParam[rA,nameCol,valCol,"krFirstFull",grkFirstFull];
rA=addParam[rA,nameCol,valCol,"krLastFull",grkLastFull];
rA=addParam[rA,nameCol,valCol,"grSDFull",grSDFull];
rA=addParam[rA,nameCol,valCol,"grZeroFull",grZeroFull];

rA=addParam[rA,nameCol,valCol,"grZeroErrFull",grZeroErrFull];
rA=addParam[rA,nameCol,valCol,"grInfFull",grInfFull];

rA=addParam[rA,nameCol,valCol,"grInfErrFull",grInfErrFull];
rA=addParam[rA,nameCol,valCol,"grTdFull",grTdFull];

rA=addParam[rA,nameCol,valCol,"grTdErrFull",grTdErrFull];

rA=addParam[rA,nameCol,valCol," "," "];
rA=addParam[rA,nameCol,valCol,"rFitData",grFitDataName];
rA=addParam[rA,nameCol,valCol,"krFirst500",grkFirst500];
rA=addParam[rA,nameCol,valCol,"krLast500",grkLast500];
rA=addParam[rA,nameCol,valCol,"grSD500",grSD500];

rA=addParam[rA,nameCol,valCol,"grZero500",grZero500];rA=addParam[rA,nameCol,valCol,"grZeroErr500",grZeroErr500];rA=addParam[rA,nameCol,valCol,"grInf500",grInf500];rA=addParam[rA

```

```

,nameCol, valCol, "grInfErr500", grInfErr500];
  rA=addParam[rA, nameCol, valCol, "grTd500", grTd500];
  rA=addParam[rA, nameCol, valCol, "grTdErr500", grTdErr500];

  rA=addParam[rA, nameCol, valCol, " ", " "];
  rA=addParam[rA, nameCol, valCol, "rFitData", grFitDataName];
  rA=addParam[rA, nameCol, valCol, "krFirst200", grkFirst200];
  rA=addParam[rA, nameCol, valCol, "krLast200", grkLast200];
  rA=addParam[rA, nameCol, valCol, "grSD200", grSD200];

rA=addParam[rA, nameCol, valCol, "grZero200", grZero200]; rA=add
Param[rA, nameCol, valCol, "grZeroErr200", grZeroErr200]; rA=add
Param[rA, nameCol, valCol, "grInf200", grInf200];

rA=addParam[rA, nameCol, valCol, "grInfErr200", grInfErr200];
  rA=addParam[rA, nameCol, valCol, "grTd200", grTd200];
  rA=addParam[rA, nameCol, valCol, "grTdErr200", grTdErr200];

  rA=addParam[rA, nameCol, valCol, " ", " "];
  rA=addParam[rA, nameCol, valCol, "rFitData", grFitDataName];
  rA=addParam[rA, nameCol, valCol, "krFirst100", grkFirst100];
  rA=addParam[rA, nameCol, valCol, "krLast100", grkLast100];
  rA=addParam[rA, nameCol, valCol, "grSD100", grSD100];
  rA=addParam[rA, nameCol, valCol, "grZero100", grZero100];

rA=addParam[rA, nameCol, valCol, "grZeroErr100", grZeroErr100];
  rA=addParam[rA, nameCol, valCol, "grInf100", grInf100];

rA=addParam[rA, nameCol, valCol, "grInfErr100", grInfErr100];
  rA=addParam[rA, nameCol, valCol, "grTd100", grTd100];
  rA=addParam[rA, nameCol, valCol, "grTdErr100", grTdErr100];

  rA=addParam[rA, nameCol, valCol, " ", " "];
  rA=addParam[rA, nameCol, valCol, "sFitData", gsFitDataName];

rA=addParam[rA, nameCol, valCol, "ksFirstFull", gskFirstFull];
  rA=addParam[rA, nameCol, valCol, "ksLastFull", gskLastFull];
  rA=addParam[rA, nameCol, valCol, "gsSDFull", gsSDFull];
  rA=addParam[rA, nameCol, valCol, "gsZeroFull", gsZeroFull];

rA=addParam[rA, nameCol, valCol, "gsZeroErrFull", gsZeroErrFull
];
  rA=addParam[rA, nameCol, valCol, "gsInfFull", gsInfFull];

rA=addParam[rA, nameCol, valCol, "gsInfErrFull", gsInfErrFull];
  rA=addParam[rA, nameCol, valCol, "gsTdFull", gsTdFull];

```

```

rA=addParam[rA,nameCol,valCol,"gsTdErrFull",gsTdErrFull];

    rA=addParam[rA,nameCol,valCol," "," "];
    rA=addParam[rA,nameCol,valCol,"sFitData",gsFitDataName];
    rA=addParam[rA,nameCol,valCol,"ksFirst500",gskFirst500];
    rA=addParam[rA,nameCol,valCol,"ksLast500",gskLast500];
    rA=addParam[rA,nameCol,valCol,"gsSD500",gsSD500];
    rA=addParam[rA,nameCol,valCol,"gsZero500",gsZero500];

rA=addParam[rA,nameCol,valCol,"gsZeroErr500",gsZeroErr500];
    rA=addParam[rA,nameCol,valCol,"gsInf500",gsInf500];

rA=addParam[rA,nameCol,valCol,"gsInfErr500",gsInfErr500];
    rA=addParam[rA,nameCol,valCol,"gsTd500",gsTd500];
    rA=addParam[rA,nameCol,valCol,"gsTdErr500",gsTdErr500];

    rA=addParam[rA,nameCol,valCol," "," "];
    rA=addParam[rA,nameCol,valCol,"sFitData",gsFitDataName];
    rA=addParam[rA,nameCol,valCol,"ksFirst200",gskFirst200];
    rA=addParam[rA,nameCol,valCol,"ksLast200",gskLast200];
    rA=addParam[rA,nameCol,valCol,"gsSD200",gsSD200];
    rA=addParam[rA,nameCol,valCol,"gsZero200",gsZero200];

rA=addParam[rA,nameCol,valCol,"gsZeroErr200",gsZeroErr200];
    rA=addParam[rA,nameCol,valCol,"gsInf200",gsInf200];

rA=addParam[rA,nameCol,valCol,"gsInfErr200",gsInfErr200];
    rA=addParam[rA,nameCol,valCol,"gsTd200",gsTd200];
    rA=addParam[rA,nameCol,valCol,"gsTdErr200",gsTdErr200];

    rA=addParam[rA,nameCol,valCol," "," "];
    rA=addParam[rA,nameCol,valCol,"sFitData",gsFitDataName];
    rA=addParam[rA,nameCol,valCol,"ksFirst100",gskFirst100];
    rA=addParam[rA,nameCol,valCol,"ksLast100",gskLast100];
    rA=addParam[rA,nameCol,valCol,"gsSD100",gsSD100];
    rA=addParam[rA,nameCol,valCol,"gsZero100",gsZero100];

rA=addParam[rA,nameCol,valCol,"gsZeroErr100",gsZeroErr100];
    rA=addParam[rA,nameCol,valCol,"gsInf100",gsInf100];

rA=addParam[rA,nameCol,valCol,"gsInfErr100",gsInfErr100];
    rA=addParam[rA,nameCol,valCol,"gsTd100",gsTd100];
    rA=addParam[rA,nameCol,valCol,"gsTdErr100",gsTdErr100];

    rA=addParam[rA,nameCol,valCol," "," "];

rA=addParam[rA,nameCol,valCol,"grrFluct12[[2]]",grrFluct12[

```

```

[2]]];

rA=addParam[rA,nameCol,valCol,"grrFluct12[[3]]",grrFluct12[
[3]]];

rA=addParam[rA,nameCol,valCol,"grrFluct12[[4]]",grrFluct12[
[4]]];

rA=addParam[rA,nameCol,valCol,"grrFluct12[[5]]",grrFluct12[
[5]]];

    rA=addParam[rA,nameCol,valCol," "," "];
    rA=addParam[rA,nameCol,valCol,"grZero3Pts",grZero3Pts];

rA=addParam[rA,nameCol,valCol,"grZeroErr3Pts",grZeroErr3Pts
];
    rA=addParam[rA,nameCol,valCol,"grInf3Pts",grInf3Pts];

rA=addParam[rA,nameCol,valCol,"grInfErr3Pts",grInfErr3Pts];
    rA=addParam[rA,nameCol,valCol,"grTd3Pts",grTd3Pts];

rA=addParam[rA,nameCol,valCol,"grTdErr3Pts",grTdErr3Pts];

rA=addParam[rA,nameCol,valCol,"log10Td3Pts)",Log10[grTd3Pts
]];
    rA=addParam[rA,nameCol,valCol,"0.43err/tau3pts)",0.434
grTdErr3Pts/grTd3Pts];
    rA=addParam[rA,nameCol,valCol,"grSD3Pts",grSD3Pts];

    rA=addParam[rA,nameCol,valCol," "," "];
    rA=addParam[rA,nameCol,valCol,"grZeroFull",grZeroFull];

rA=addParam[rA,nameCol,valCol,"grZeroErrFull",grZeroErrFull
];
    rA=addParam[rA,nameCol,valCol,"grInfFull",grInfFull];

rA=addParam[rA,nameCol,valCol,"grInfErrFull",grInfErrFull];
    rA=addParam[rA,nameCol,valCol,"grTdFull",grTdFull];

rA=addParam[rA,nameCol,valCol,"grTdErrFull",grTdErrFull];

rA=addParam[rA,nameCol,valCol,"log10grTdAll",Log10[grTdFull
]];
    rA=addParam[rA,nameCol,valCol,"log10grTdErFull",0.434
grTdErrFull/grTdFull];
    rA=addParam[rA,nameCol,valCol,"grSDFull",grSDFull];

```

```

rA=addParam[rA,nameCol,valCol," "," "];
rA=addParam[rA,nameCol,valCol,"nParams",nParams];

rA=addParam[rA,nameCol,valCol,"NotebookFileName",NotebookFi
leName[]];

(*rA//MatrixForm*)
(*Print[Length[rA]];
Print[rA[[35]]];
Abort[];*)

Do[(*i down*)
  jLast=Length[rA[[i]]];
  Do[(*j across*)
    WriteString[res,ToString[ff[rA[[i,j]]]]];

If[j!=jLast,WriteString[res,","],WriteString[res,"\n"]]
  ,{j,1,jLast}
  ]
  ,{i,Length[rA]}
  ];
Close[res];
Print["results filename= ",resultsFileName];
restIPrint[]
);
(*saveResults[];*)

(*****
*****
(* combine all graphics and export
*)
(*****
*****
ClearAll[savePlots];

savePlots[]:=
(
  saveIPrint[0];
  ClearAll[pageLabel];

(*bs(*BaseStyle*)={FontFamily["CourierNew",FontSize[12];*)
pageLabel=
  Framed[
    Graphics[
      Inset[
        Pane[

```

```

        TextCell[
            StringJoin[inFileName, " =
inFileName","\n",ToString[ iDot ] ," = iDot" ,
"\n",FileName[NotebookFileName[]]," =
notebook","\n",dateString," = date","\n", timeString, " =
time"],
            "Text",
            FontSize->10
        ],
        144 (*pane width pts*)
    ],
    {0,0} (*inset pos*)
],
ImageSize->{144,108}
];
allPlots=GraphicsGrid[{
    {pageLabel,vhEffPlot,bkgEffPlot, bkgCorrPlot},
    {vchcPlot,vchcDiffPlot,sPlot,rPlot},
    {gsrFullPlot, gsr500Plot ,gsr200Plot, gsr100Plot},
    {grsFullPlot,grs500Plot,grs200Plot,grs100Plot},
    {gsFitPlotFull,gsFitPlot500, gsFitPlot200,
gsFitPlot100},
    {grFitPlotFull,grFitPlot500, grFitPlot200,
grFitPlot100}
}, ImageMargins->{{36,36},{36,36}}];
plotFileName=base<>"-"<>"plots"<>"-
"<>timeStampString<>".pdf";
Export [plotFileName, allPlots];
Print["plot filename= ",plotFileName];
restIPrint[];
);
(*savePlots[]*)

(*****
*****
(* show analysis done this dot
*)
(*****
*****
ClearAll[done];
done[]:=Print["dot ", iDot, " processed and products
saved\n\n"];
(*done[]*)

(*****
*****

```

```

*****
(* process one dot completely
*)
(*****
*****
ClearAll[processOneDot];
processOneDot[]:=
  (optimize[];
  showOptResults[];

fitAnisDecay[];fitAnisDecay3Pts[];printAnisFitNums[];fitInt
enDecay[];
  saveResults[];
  If[isSavePlots,savePlots[]];
  done[]);
(*processOneDot[]*)

(*****
**)
(*          process dots this file
*)
(*          iDot=-1 □ no background traces
*)
(*****
**)
ClearAll[dotsThisFile];
dotsThisFile[(*inFileName_,iFirstDot_,iLastDot_*)]:=Module[
{(*inFileName*)(*,iFirstRow,isBkgs,data,dim,iLastRow,nCols,
nDots,n,iDot,cBkg,iTime,iv,ih,ibv,ibh,ibTime,v,h,bv,bh,bvf,
bhf,gbv,gbh,vhRawPlot,bkgRawPlot,
bkgCorrPlot,inputPlot,s,dep*)},
  saveIPrint[];

  (* generate timestamp for all output this pass *)
  isStamp=True;
  tdList=DateList[];
  dateFormat={"Year","-","Month","-","Day"};
  timeFormat={"Hour",":","Minute",":","Second"};
  dateString=DateString[tdList,dateFormat];
  timeString=DateString[tdList,timeFormat];
  timeStampFormat={(*"-",*)"Year",(*"-",*)"Month",(*"-",
*,*)"Day",*"-","Hour24","Minute","Second"};

timeStampString=If[isStamp,DateString[tdList,timeStampForma
t],""];
  (*dP[timeStampString];*)

```

```

(*inFileName="IgE-1-LRB.csv";
q=DialogInput[{},Column[

{"inFileName=\t"InputField[Dynamic[inFileName],String,Align
ment☐Right],

(*"isBkgs=\t"InputField[Dynamic[isBkgs],Number,Alignment☐Ri
ght],

"cbkgs=\t"InputField[Dynamic[cbkg],Number,Alignment☐Right],

"iFirst=\t"InputField[Dynamic[iFirstRow],Number,Alignment☐R
ight],*)
  Button["proceed",DialogReturn[{{inFileName,
isBkgs,cbkg,iFirst}}]]];
  inFileName=q[[1]];*) (* dP[inFileName];*)

  data=Import[inFileName];(*dP[inFileName];*)
  dim=Dimensions[data]; (*dP[dim];*)
  len=dim[[1]];(*dP[len];*)
  nCols=dim[[2]]; (*dP[nCols];*)
  If[Mod[nCols,3]!=0, Print["nCols not divisible by
3"];Exit[]];
  nDots=(nCols-6)/3; (*dP[nDots];*)
  (*nDots=1;*)
  n(*num pts*)=len-iFirst +1; (*dP[n];*)
  i1=If[iFirstDot<1, 1,iFirstDot];
  i2=If[iLastDot<1, nDots,iLastDot];

  For [iDot=i1,iDot<=i2,iDot++,
    (*Print info this dot*)
    Print["Analyzing ",inFileName," dot=", iDot, " of ",
nDots];
    Print["Model: ",model];
    base= StringTake[inFileName, StringLength[inFileName]-
4]<>If[iDot!=-1,"-d"<>IntegerString[iDot,10,2],""];
    ibTime=1;ibv=2;ibh=3;
    iTime=4+3*iDot;iv=5+3*iDot;ih=6+3*iDot;
    dataRange={};
    time=data[[iFirst;;len,iTime]]; (*1st dim is DOWN, 2nd
ACROSS *)
    vEff=data[[iFirst;;len,iv]]; (*starts 2nd row,
indicated col *)
    hEff=data[[iFirst;;len,ih]]; (*starts 2nd row,
indicated col *)
    If [iDot!=-1, (* yes, bkgs *)
      bvEff=data[[iFirst;;len,ibv]]; (*starts 2nd row,

```

```

indicated col *)
    bhEff=data[[iFirst;;len,ibh]],
    (*else*)
    bvEff=Table[0,{n}];
    bhEff=Table[0,{n}]
];(*if*)

    bvf=bvEff-Mean[bvEff]; bhf=bhEff-Mean[bhEff];
    gbv=dataCorrelate[bvf,bvf];
    gbh=dataCorrelate[bhf,bhf];
    range={0, 1.25Max[vEff,hEff]};
    inputPlots={
        vhEffPlot=ListPlot[{vEff,hEff},PlotLabel-
>"vhEff",PlotRange->range],
        bkgEffPlot=ListPlot[{bvEff,bhEff},PlotLabel-
>"bkgRaw",PlotRange-
>range],bkgCorrPlot=ListPlot[{gbv,gbh},PlotLabel-
>"bkgCorr"]];
    ListPlot[{gbv,gbh},PlotLabel->"bkgCorr"];
    processOneDot[]
    ] ;(*for*)
    restIPrint[];(* ";" needed here; otherwise prints "0"*)
]; (*module*)
(*inFileName="IgE-1-LRB.csv";dotsThisFile[];*)

(*****
*****
*)
    process all dots all files
*)
(* run the code ABOVE once before running this, possibly
multiple times *)
(*****
*****
)

ClearAll[dotsAllFiles];
iWeight=2; (* 1=uniform wt, 2=s^2 wt; 3=full s,r pt wt*)
nParams=2; (*2 or 3 p *)
tauMin=0.1; (* min tau allowed*)
isSavePlots=True; (* "False" speeds up checking calcn's &
saves disc space *)

dotsAllFiles[]:=
(fileList=getFileList[];
nFiles=Length[fileList];
iFirstFile=1; iLastFile=1;
If[iLastFile==0, iLastFile=nFiles];

```

```

iFirstDot=1;iLastDot=2;
For[i=iFirstFile,i<=iLastFile,i++,
  inFileName=fileList[[i]];
  (*inFileName="Pf-1-LRB-4.csv"; (* REMOVE THIS to
analyze all QD in arb directory *)*)
  inFileName="IgE-1-LRB.csv"; (* REMOVE THIS to analyze
all QD in arb directory *)
  (*inFileName="2015-04-15-para-10pl-LRB.csv"; (* REMOVE
THIS to analyze all QD in arb directory *)*)
  model="v140-2wt-2p-pltau" ;(* THIS MUST BE FILLED IN
BEFORE STARTING ANALYSIS *)
  dotsThisFile[]
  ]);
dotsAllFiles[]

```

Program Code 4: MergeFiles v.39

```

(*****
*****
(* MERGEFILES
*)
(* Merge and avg "results" CSV files from multiple QDs
*)
(* mode=0 → identifies and avg's "results" files into

```

```

"nums.csv" *)
(* mode=1 → avgs only cols in merged "nums.csv" w/ non-
blank 1st cells *)
(* mode=2 → replaces "nums/csv" with saved "numsSaved.csv"
to allow new mode 1 *)

(* USE:
*)
(* 1) run mode 0 to process and avg ALL files in curr
directory into 'nums.csv' *)
(* 2) in "nums.csv" edit "" into 1st cells of QD to be
removed; run mode 1 *)
(* 3) to get back orig 'nums.csv' with ALL QD, run mode 2
*)

(* NOTE:
*)
(* This prog intended ONLY for use with "results" files
created by *)
(* RS-sWtd-485-FINALcopy-mod15k.nb
*)
(* and re-processed by NoGrrInf-17-New3PtFit.nb
*)
(*****
*****

(*****
)
(* initialize Mathematica system
*)
(*****
)
ClearSystemCache[];
ClearAll
[iFirstRow, iTime, iv, ih, dP, aP, se, uGeom, wGeom, uMean, wMean, mer
geFiles, i];
SetDirectory[NotebookDirectory[]];

Needs["CCompilerDriver`"];
Needs["SymbolicC`"];
Needs["CompiledFunctionTools`"];
Unprotect[CompiledFunctionools`Private`getInstruction];
CompiledFunctionTools`Private`getInstruction[line_, {0, _}] :=
CompiledFunctionTools`Private`getInstruction[line, {3, 1}]
SetDirectory[NotebookDirectory[]];
(*SetOptions[$FrontEndSession, PrintingStyleEnvironment<>"Wor

```

```

king"];*)
Off[General::munfl];
gPrint= If[iPrint!=0,Print[##]]&;
iPrint=1;
one=#1&;
remOdd[ e_, x_]:=((e/.{x->Power[x,2]})/.{ x2->0, x6->0, x10->
0,x14-> 0})/.{x-> x1/2};
SetAttributes[dP,HoldAll];
dP[z_]:=If[(iPrint!=0 &&ValueQ[z]==True)||
(ValueQ[iPrint]==False),Print[HoldForm[z],"=",ReleaseHold[z
](*//MatrixForm*)]]; (* debug print*)
aP[a_]:=Module[{last},last=Min[8, Length[a]];a[[1;;last]]];
se[x_]:=Simplify[Expand[x]]; (* simplify-expand *)

(*****
*****
(*getFileList[]:  get names of raw data files in current
directory *)
(*****
*****
ClearAll[getFileList];
getFileList[]:=Module[({*f,l,c1,c2,c3,c4,c,i*}),
  f=FileNames[All];l=Length[f];
  c2=StringContainsQ[f, "csv"];
  c3=Thread[StringContainsQ[f, "results"]];
  c=Table[c2[[i]]&&c3[[i]],{i,l}];
  fs={};For [i=1,i<=l,i++,
    If[c[[i]],AppendTo[fs,f[[i]]]];
  Return[fs]
];

(*****
*****
(* "_means" :  define various averaging modes
*)
(*****
*****

ClearAll[uMean,wMean,uGeom,wGeom];
uMean[x_,sd_,ok_]:=Module[{nDots,w,wTot(*w*)}, (* simple
arith avg/sd *)
  (*iu=iii;dP[iu];*)
  (*Print["uMean"];*)
  nDots=Length[x];
  nUsed=nDots;
  w=ok; (* here, w is integer = 0 or 1 *)(*dP[ok];*)
  wTot=Total[w];(*dP[wTot];*)

```

```

wAvg=Total[x*w]/wTot; (* dP[wAvg]; *)
wSd=(Total[(x-wAvg)^2*w]/wTot)^(1/2)*sqrt(nUsed/(nUsed-1)) ;
Return[{wAvg,wSd}]];
wMean[x_,sd_,ok_]:=Module[{j,nDots,nUsed,w,wTot,wN}, (*wtd
arith avg*)
(*iw=iii;dP[iw];*)
(*Print["wMean"];*)
ClearAll[j];
nDots=Length[x]; (*ok=1 or ok=0*)
w=Table[If[sd[[j]]>2^20,0,
If[sd[[j]]>0,ok[[j]]/sd[[j]]^2,0]],{j,nDots}];
nUsed=Sum[If[w[[j]]>0,1,0],{j,nDots}];
wTot=Total[w];
wAvg=Total[x*w]/wTot;
wSd=(Total[(x-wAvg)^2*w]/wTot)^(1/2)*sqrt(nUsed/(nUsed-1)) ;
Return[{wAvg,wSd}]];
wGeom[
x_,sd_,ok_]:=Module[{j,nDots,w,nUsed,wTot,wN,xn,lx,avgLx}, (*
*wtd geom avg*)
(*ig=iii;dP[ig];*)
(*Print["gMean"];*)
nDots=Length[x];
w=Table[If[sd[[j]]>0
&& x[[j]]>0,ok[[j]]*x[[j]]^2/sd[[j]]^2,0],{j,nDots}];
nUsed=Sum[If[w[[j]]>0,1,0],{j,nDots}];
wTot=Total[w];
lx=Table[If[x[[i]]>0,Log[x[[i]]],0],{i,nDots}];
lxAvg=Total[lx*w]/wTot; (*dP[avgLx];*)
gAvg=Exp[lxAvg]; (* wtd geom avg *)
lVar=(Total[(lx-lxAvg)^2*w]/wTot*nUsed/(nUsed-1))^1;
lSD=lVar^(1/2);
gSd=gAvg*(E^lSD-1);
(*dP[{gAvg,gSd,lSD}];*)
Return[{gAvg,gSd,lSD}]];

(*****
*****
(* actual prog "MergeFiles" begins here
*)
(*****
*****

fileName="paramListFile.csv";
fileListFileName="fileListFile.csv";
inFileName="inFile.csv";

```

```

outFileName="nums.csv";
savedOutFileName="numsSaved.csv";

iFirstDotColumn=5; (* cols name, value, blank, blank
precede *)
ClearAll[mergeFiles];
mergeFiles[iModeTemp_, paramListFileName_, outFileName_,
savedOutFileName_] :=
  Module[{i},
    (* iMode 0 □ combine & avg into "nums.csv" AND(!) copy
"nums.csv" into "numsSaved.csv" *)
    (* iMode 1 □ re-average "numsSaved.csv" with selected
columns omitted into "nums.csv"*)
    (* iMode 2 □ copy "numsSaved.csv" into "nums.csv" with
selected columns omitted *)

    (* get iMode this pass *)
    iMode=Input["mode (0, 1 or 2) = ",iModeTemp];
    Print["iMode=", iMode];

    (* continue *)
    If [iMode==0 ,
      files=getFileList[];dP[files];
      nFiles=Dimensions[files][[1]];dP[nFiles];
      (*nFiles=16;*)
      (*For [ i=iFirstDotColumn, i<=iFirstDotColumn+nFiles-
1,i++,*)
      For [i=1, i<=nFiles,i++,
        currFile=files[[i]];(*dP[currFile];*)
        Open[currFile];
        data=Import[currFile];
        (*dP[Dimensions[data]];dP[[data]];*)
        mCols=Dimensions[data][[2]];
        iLastDotColumn=nFiles+iFirstDotColumn -
1;(*dP[iLastDotColumn];*)
        If[!FileExistsQ[paramListFileName],Print["no
paramListFile; exiting"];Abort[]];
        If [i==1,
          paramListFile=Import[paramListFileName];
          mRows= Count[paramListFile,Except[""]];
          params=ArrayReshape[paramListFile,{mRows,1}];
          (*dP[Dimensions[params]];*) (*dP[params];*)
          blanks=Table["",{mRows},{iFirstDotColumn-2}];
          (*dP[Dimensions[blanks]];*)
          outFile=Join[params, blanks,2]; (*
dP[Dimensions[outFile]];*)
          outFile[[1,2]]="avg";

```

```

    outFile[[1,3]]="avg stderr";
    ];(*if i=1*)
WriteString["stdout",ToString[i],"-"];
If[i>1 && Mod[i,32]==0,WriteString["stdout", "\n"]];
ones=Table[1,mRows];
outFile=Join[outFile,data[[1;;mRows,2;;2]],2];
]; (*for*)
];(*if0*)
(*dP[nFiles];
Export["x.csv", outFile];
gPrint["end if i=0"];
Abort[];*)

(*dP[iMode];*)
If [(iMode == 1|| iMode==2) ,
    If[iMode==1,If[!FileExistsQ[outFileName],Print["no
'outFile' to re-avg; exiting"];Abort[],
        outFile=Import[outFileName]]];
    If[iMode==2,If[!FileExistsQ[savedOutFileName],Print["no
'savedOutFile' to re-avg; exiting"];Abort[],
        outFile=Import[savedOutFileName]]];
    (*Print["in loop"];*)
    mRows=Dimensions[outFile][[1]];
    mCols=Dimensions[outFile][[2]];
    nFiles=mCols-iFirstDotColumn+1;
    iLastDotColumn=mCols;
    (*outFile=avgInFile;*)
    ]; (*if*)
    (*gPrint["paused"];
    Pause[1*^9];*)

(*all of iMode=0 and =1 and =2 come here*)
iLastDotColumn=iFirstDotColumn+nFiles-1;
dP[iFirstDotColumn];dP[iLastDotColumn];
ok=Table[If[outFile[[1,i]]=="value",1,0],{i,
iFirstDotColumn,iLastDotColumn}];dP[ok];
If[iMode==2,
    For[i=iFirstDotColumn,i<=iLastDotColumn,i++, (*sheet
index i*)
        ii=i-iFirstDotColumn+1; (*file index ii*)
        Print["i=",i,"  ii=",ii];
        gMin=.12;gMax=.55;
        If[(outFile[[25,i]]<gMin)|| (outFile[[25,i]]>gMax),
ok[[ii]]=0;outFile[[1,i]]=""];

        gr0Min=-.1;gr0Max=.1;
        If[(outFile[[34,i]]<gr0Min)|| (outFile[[34,i]]>gr0Max),

```

```

ok[[ii]]=0;outFile[[1,i]]="";
  If[(outFile[[45,i]]<gr0Min)|| (outFile[[45,i]]>gr0Max),
ok[[ii]]=0;outFile[[1,i]]="";
  If[(outFile[[56,i]]<gr0Min)|| (outFile[[56,i]]>gr0Max),
ok[[ii]]=0;outFile[[1,i]]="";
  If[(outFile[[67,i]]<gr0Min)|| (outFile[[67,i]]>gr0Max),
ok[[ii]]=0;outFile[[1,i]]="";

  grTdMin=0.;grTdMax=500.;

If[(outFile[[38,i]]<grTdMin)|| (outFile[[38,i]]>grTdMax),
ok[[ii]]=0;outFile[[1,i]]="";

If[(outFile[[49,i]]<grTdMin)|| (outFile[[49,i]]>grTdMax),
ok[[ii]]=0;outFile[[1,i]]="";

If[(outFile[[60,i]]<grTdMin)|| (outFile[[60,i]]>grTdMax),
ok[[ii]]=0;outFile[[1,i]]="";

If[(outFile[[71,i]]<grTdMin)|| (outFile[[71,i]]>grTdMax),
ok[[ii]]=0;outFile[[1,i]]="";

  ](*for*)
];(*if*)

uMeanTargets={17,18,19,20,21,22,23,24,25,26,27,28,33,44,55,
66,77,88,99,110,118,119,120,121,131,141};
wMeanTargets={34,36,45,47,56,58,
67,69,78,80,89,91,100,102, 111,113, 123,125,133,135};
wGeomTargets={38,49,60,71,82,93,104,115,127,137};
log10Tau={129,139};
log10TauErr={130,140};

(*dP[nFiles];*)
iLastDotColumn=iFirstDotColumn+nFiles-1;
outFile[[All,2;;3]]="";(*clear avg, err cols*)
dP[mRows];
dP[Dimensions[outFile]];
For[i=1,i<=mRows,i++,
  iii=i;
  (*dP[i];*)
  o= outFile[[i,iFirstDotColumn;;iLastDotColumn]];(*
dP[o];*)

op1=If[i<mRows,outFile[[i+1,iFirstDotColumn;;iLastDotColumn
]],Table[1,{nFiles}]];

```

```

op1Tot=Total[op1];
If[MemberQ[uMeanTargets,i],
  avg=uMean[o,op1,ok];
  outFile[[i,2]]=avg[[1]];
  outFile[[i,3]]=avg[[2]]];
If[MemberQ[wMeanTargets,i]&&op1Tot>0,
  avg=wMean[o,op1,ok];(* dP[o];dP[op1];dP[ok];*)
  outFile[[i,2]]=avg[[1]];
  outFile[[i,3]]=avg[[2]]];
If[MemberQ[wGeomTargets,i]&&op1Tot>0,
  avg=wGeom[o,op1,ok];
  outFile[[i,2]]=avg[[1]];
  outFile[[i,3]]=avg[[2]]];

]; (* for *)

For[i=1,i<=mRows,i++, (*dP[i];*) (* seems to be
circular ref though how ??? *)
  If[MemberQ[log10Tau,i],
    outFile[[i,2]]=Log10[outFile[[i-2,2]]];
    (*outFile[[i,3]]=0.434 avg[[3]]);
dP[outFile[[i,3]]];*)
    outFile[[i,3]]=Log10[E]* Log[1+outFile[[i-
2,3]]/outFile[[i-2,2]]];
  ]; (*if*)
]; (* for*)

Print["dim[outFile]=" ,Dimensions[outFile]];
If[FileExistsQ[outFileName], DeleteFile[outFileName]];
outFile[[1,2]]="avg";
outFile[[1,3]]="st dev";
outFile[[1,4]]="(blank col)";
Export[outFileName,outFile];
If [iMode==0,
  If[FileExistsQ[savedOutFileName],
DeleteFile[savedOutFileName]];
  CopyFile[outFileName, savedOutFileName]];
(*avgInFile=outFile;*)
];

mergeFiles[0,"paramNamesColumnNew.csv",
"nums.csv","numsSaved.csv" ];

iMode= 0
files = {2015-04-15-para-10pl-LRB-d01-results-20201001-
134712.csv,2015-04-15-para-10pl-LRB-d02-results-20201001-
134719.csv,2015-04-15-para-10pl-LRB-d03-results-20201001-

```

```
134726.csv,2015-04-15-para-10pl-LRB-d04-results-20201001-
134734.csv,2015-04-15-para-10pl-LRB-d05-results-20201001-
134741.csv,2015-04-15-para-10pl-LRB-d06-results-20201001-
134748.csv}
nFiles = 6
1-2-3-4-5-6-
iFirstDotColumn = 5
iLastDotColumn = 10
ok = {1,1,1,1,1,1}
mRows = 144
Dimensions[outFile] = {144,10}
iu = 17
iu = 18
iu = 19
iu = 20
iu = 21
iu = 22
iu = 23
iu = 24
iu = 25
iu = 26
iu = 27
iu = 28
iu = 33
iw = 34
iw = 36
ig = 38
iu = 44
iw = 45
iw = 47
ig = 49
iu = 55
iw = 56
iw = 58
ig = 60
iu = 66
iw = 67
iw = 69
ig = 71
iu = 77
iw = 78
iw = 80
ig = 82
iu = 88
iw = 89
iw = 91
ig = 93
```

```

iu = 99
iw = 100
iw = 102
ig = 104
iu = 110
iw = 111
iw = 113
ig = 115
iu = 118
iu = 119
iu = 120
iu = 121
iw = 123
iw = 125
ig = 127
iu = 131
iw = 133
iw = 135
ig = 137
iu = 141
dim[outFile]= {144,10}
avg
{46.0766,178.559,1.58417}

```

```

ClearAll[x,sd,w,ok,lAvg];
nDots=16;
x={16.37268197, 7.050827591,3.389313532,
242.4241185,465.5862344 ,26.24850047 ,7.382614208
,10.33169337 ,6.119239947 ,17.52866473 ,21.51468385
,38.816786 ,12.09501695 ,15.39400941 ,65.26227094
,5.888714016};
sd={2.342689162 ,0.771996157, 0.321129931, 8.550285455
,264.0147962, 3.486522097, 0.886088484, 1.084189108,
0.920431403, 4.531403416,1.788593673, 3.81401505,
1.326592628 ,1.732702052, 4.904354864 ,0.626745194};
ok={1,1,1,1,1,1,1,1,1,1,1,1,1,1,1};

w=Table[If[sd[[j]]>0
&& x[[j]]>0,ok[[j]]*x[[j]]^2/sd[[j]]^2,0},{j,nDots}];dP[w];
lx=Log[x];dP[lx];
lAvg=Total[lx*w]/Total[w];dP[lAvg];
avg=ElAvg;

lVar=Total[(lx-lAvg)^2 w]/Total[w]*nDots / (nDots-1);dP[lVar];

```

```

lSD= $\sqrt{lVar}$  ;dP[lSD];

lSdFact=(ElSD-1);dP[lSdFact];
e=avg*lSdFact;dP[e];
q=wGeom[x, sd, ok];dP[q];
dl=Log10[E]*q[[3]]

w =
{48.8439, 83.4161, 111.394, 803.879, 3.10988, 56.6793, 69.4171, 90
.8098, 44.199, 14.9635, 144.693, 103.58, 83.1262, 78.9325, 177.076
, 88.2793}
lx =
{2.79561, 1.95314, 1.22063, 5.49069, 6.1433, 3.26761, 1.99913, 2.3
3522, 1.81144, 2.86384, 3.06874, 3.65885, 2.49279, 2.73398, 4.1784
1, 1.77304}
lAvg = 3.83031
lVar = 2.50961
lSD = 1.58417
lSdFact = 3.87526
e = 178.559
q = {46.0766, 178.559, 1.58417}
0.687998
lSD = 1.58417

lSdFact = 3.87526
Total[lWtd]/Total[w] = 2.35276
e = 178.559
avg
{46.0766, 178.559, 1.58417}
□

```

Program Code 5: PC Corr v.80

```

(*****
*)
(*****
*)
(*****
*)

```

```

(* Enderlein correlation from hv arrival times *)
(*****
*)
(*****
*)
(*****
*)

(*****
*)
(*Initialize *)
(*****
*)
ClearSystemCache[];
ClearAll[];
SetDirectory[NotebookDirectory[]];

(*****
*)
(* create arrays for each channel from files
*)
(*****
*)
ClearAll[makeChannelArrays];
makeChannelArrays[inFileName_]:=Module[{i1,i2,l,l1,l2,gEst,
t1,t2,w1,w2,w1Tot,w2Tot,data,psPerPt,secPerPt,maxInitPts,gT
radEst,maxTimeSec, r1,r2, gGbEst,wm,im},
  (* import data; get length; make empty tables for times,
wts*)
  secPerPt=164.61*^-12;(*DPC230 psPerPt=164.61;*)
  data=Import [inFileName] (*[[All,1;;2]]*);
  l=Length[data];
  i1=0;i2=0;im=0;
t1=Table[0,{l}];t2=Table[0,{l}];w1=Table[0,{l}];w2=Table[0,
{1}];wm=Table[0,{l}];
  tm=data[[All,1]];

  (*read data & make times, wts each ch*)
  If [Length[data[[1]]]==2,
    (*no wt column*)
    Do[
      im++;
      If[data[[i,2]]==1,i1++;
t1[[i1]]=t1[[i1]]+data[[i,1]];w1[[i1]]=N[1];wm[[im]]=N[1]];
      If[data[[i,2]]==2,i2++;
t2[[i2]]=t2[[i2]]+data[[i,1]];w2[[i2]]=N[1];wm[[im]]=N[-
1]],

```

```

        {i,1}
    ],
    (*exists wt column*)
    Do[
        im++;
        If[data[[i,2]]==1,i1++;
t1[[i1]]=t1[[i1]]+N[data[[i,1]]];w1[[i1]]=data[[i,3]]];
        If[data[[i,2]]==2,i2++;
t2[[i2]]=t2[[i2]]+N[data[[i,1]]];w2[[i2]]=data[[i,3]]],
        {i,1}
    ]
];
(* make output *)
t1=t1[[1;;i1]]; (* hv arrival times array *)
t2=t2[[1;;i2]];
l1=Length[t1]; (* length of time array - typ num hv *)
l2=Length[t2];
w1=w1[[1;;i1]]; (* wt per each time - typically 1 *)
w2=w2[[1;;i2]];
w1Tot=Total[w1]; (* tot wts or hv this channel *)
w2Tot=Total[w2];
Print[w1Tot," ",w2Tot];
latestPt=Max[t1[[l1]],t2[[l2]]];
maxTimeSec=latestPt*secPerPt;
r1=l1/maxTimeSec; (* hv per sec *)
r2=l2/maxTimeSec;
gTradEst=N[w1Tot/w2Tot];
gGbEst=N[(w1Tot-w2Tot)/(w1Tot+w2Tot)];

Print["len tot=",l];
Print["len 1=",l1];
Print["len 2=",l2];
Print["w 1=",w1Tot];
Print["w 2=",w2Tot];
Print["maxInitPts=",maxInitPts];
Print["max Time=",maxTimeSec];
Print["r1=",r1];
Print["r2=",r2];
Print["gTradEst=",gTradEst];
Print["gGbEst=",gGbEst];

(*Return[{t1,t2,l1,l2,w1,w2,gTradEst,maxTimeSec, r1,r2,
gGbEst,maxInitPts,tm,wm}];*)
Return[{t1,t2,tm,l1,l2,w1,w2,wm,latestPt,maxTimeSec,
r1,r2, gTradEst,gGbEst}]
];
(*makeChannelArrays["outRaw-1k.csv"];

```

```

Print["i1=",i1," ", "i2=",i2];
Print["t1=",t1[[1;;16]]];
Print["t2=",t2[[1;;16]]];
Print["tm=",tm[[1;;16]]];
Print["w1=",w1[[1;;16]]];
Print["w2=",w2[[1;;16]]];
Print["wm=",wm[[1;;16]]];*)

(*****
*)
(* merge two single-photon channel arrays
*)
(*****
*)
ClearAll[mergePhotonArrays];
mergePhotonArrays[a_,b_]:=Module[{},
  jaMax=Length[a];
  jbMax=Length[b];
  m=Table[0,jaMax+jbMax];

  ja=0;
  jb=1;
  i=1;
  While[True,
    bb=b[[jb]];
    (*wbb=wb[[jb]];*)
    Print[];Print["jb=",jb," bb=",bb];
    While[True,
      If[ja<jaMax,ja++,Goto[done]];
      (*Print["ja=",ja," a=",a[[ja]]];*)
      If[(aa= a[[ja]])< bb,m[[i]]=aa;i++,Break[]]
    ];(* end while *)
    Print[];Print["ja=",ja," aa=",aa];
    ja++;
    aa=a[[ja]];
    (*waa=wa[[ja]];*)
    While[True,
      If[jb<jbMax,Goto[done]];
      Print["jb=",jb," bb=",b[[jb]]];
      If[(bb=b[[jb]])<aa,m[[i]]=bb;i++,Break[]];
      jb++;
    ] (* end while *)
  ];(* end while *)
  Label[done];
  Return[m]
];
(*a={10,30,50,60,90,120,150,190};

```

```

b={11,15,45,71,80,93,110,135};
jaMax;
jbMax;m
mergePhotonArrays[a,b]*)

(*****
*)
(* enderlein deflate list of hv times
*)
(*****
*)
ClearAll[deflate];
deflate[x_,w_,j_]:=Module[{l,i,xx,xxTemp,ww,wwTemp,wwTerm,i
New},
  (* divide data times by pwr of 2 *)
  l=Length[x];
  If[l<2, Break[]];
  fact=2j;
  xx=IntegerPart[x/fact];
  ww=w;

  (* combine new pts at equal time *)
  For [i=2,i<=l,i++,
    If[ xx[[i]]==xx[[i-1]],
      ww[[i]]=ww[[i]]+ww[[i-1]];
      ww[[i-1]]=0;
      xx[[i-1]]=0;
    ];
  ];

  (* remove 0-wt points *)
  xxTemp=Table[0,l];
  wwTemp=xxTemp;

  iNew=0;
  For [i=1,i<= l,i++,
    If[ (wwTerm=ww[[i]])!=0,
      iNew++;
      wwTemp[[iNew]]=wwTerm;
      xxTemp[[iNew]]=xx[[i]]
    ];
  ];

  Return[{xxTemp[[1;;iNew]],wwTemp[[1;;iNew]]}]
];
(*Print["x=",x={1,10,20,30,32,34,40,50}];
Print["wx=",wx={1,1,1,3,7,1,1,1}];

```

```

Print["x=",deflate[x,wx,2][[1]]];
Print["wx=",deflate[x,wx,2][[2]]];
Print["wtMax=", deflate[x,wx,2][[3]]];
Print["tot wx=", Total[deflate[x,wx,2][[2]]];*)

(*****
*)
(* enderlein 'deflate' - PARALLEL
*)
(* about 3.3 sec(TS 0)-4.6 sec (TS 30)/546031 pts
*)
(* .085 sec / 10k pts
*)
(*****
*)
Needs["CompiledFunctionTools`"];
ClearAll[deflateCompile];
deflateCompile=Compile[
  {{x,_Integer,1},
   {w,_Real,1},
   {j,_Integer}},

Module[{done,l,i,xx,xxTemp,ww,wwTemp,wwTerm,iNew,fact,qxx,q
ww,q},
  l=Length[x];
  If[l<2, Goto[done]];

  fact=2j;
  xx=IntegerPart[x/fact];
  ww=w;
  For [i=2,i<=l,i++,
    If[ xx[[i]]==xx[[i-1]],
      ww[[i]]=ww[[i]]+ww[[i-1]];
      ww[[i-1]]=0.0;
      xx[[i-1]]=0;
    ];
  ];

  xxTemp=Table[0,{l}];
  wwTemp=Table[0.,{l}];
  iNew=0;

  For [i=1,i<= l,i++,
    wwTerm=ww[[i]];
    If[wwTerm==0,Goto[endIf2]];
    iNew++;

```

```

wwTemp[[iNew]]=wwTerm;
xxTemp[[iNew]]=xx[[i]];
Label[endIf2];
];

Label[done];
q={xxTemp[[1;;iNew]],wwTemp[[1;;iNew]]};
Return[q]
],
RuntimeAttributes->{Listable},Parallelization->True];

(*CompilePrint[deflateCompile]
Print[(* /n*) "x=",x={1,10,20,30,32,34,40,50}];
Print["wx=",wx=N[{1,1,3,7,1,1,1,1}]];
Print["x=",deflateCompile[x,wx,2][[1]]];
Print["w=",deflateCompile[x,wx,2][[2]]];
Print["tot wx=", Total[deflateCompile[x,wx,2][[2]]]];*)

(*****
*)
(* enderlein single-photon correlation
*)
(*****
*)
ClearAll[endCorr];
endCorr[a_,b_,wa_,wb_,tau_]:=
Module[{bShift,g,jStart,jMax,
ja,jb,jaMax,jbMax,aa,bb,iMax,iPass,maxPass,wbb,bLast,sa,sb,
a1,waa,s2ab,sab,na,nb,s2g},

jaMax=Length[a]; jbMax=Length[b];jMax=Max[jaMax,jbMax];
(*num times or bins*)
g=0;

(*Print["entering endCorr...."," iOrder=", iOrder, "
tau=",tau];
iMax=Min[32,Length[a]];
Print["a=",a[[1;;iMax]]];
Print["wa=",wa[[1;;iMax]]];
Print ["jaMax=",jaMax,"  jbMax=",jbMax(*, "
jMax=",jMax*);
Print ["ta=",ta,"  tb=",tb, "  tMax=",tMax];
Print ["na=",na,"  nb=",nb];
Print ["sa=",sa,"  sb=",sb,"  sab=",sab];
Print ["var wa=",Variance[wa],"  var wb=",Variance[wb]];
Print["entering 'ab'...."];*)

```

```

ja=0;
jb=1;
bShift=b+tau;
While[True,
  bb=bShift[[jb]];
  wbb=wb[[jb]];
  (*Print[];Print["jb=",jb,"  bb=",bb];*)
  While[True,
    If[ja<jamax,ja++,Goto[done]];
    (*Print["ja=",ja,"  a=",a[[ja]];*)
    If[(aa= a[[ja]])>=
bb,If[aa==bb,p=wa[[ja]]*wbb;g=g+p];Break[]
  ];(* end while *)
  (*Print[];Print["ja=",ja,"  aa=",aa]; *)
  aa=a[[ja]];
  waa=wa[[ja]];
  While[True,
    If[jb<jbMax,jb++,Goto[done]];
    (*Print["jb=",jb,"  bb=",bShift[[jb]];*)
    If[(bb=bShift[[jb]])>=aa, If[aa==bb,p=waa*wb[[jb]];
g=g+p];Break[]
  ] (* end while *)
];(* end while *)

Label[done];
Return[N[g]]
];(* end module 'endCorr' *)

(*****
*)
(* enderlein single-photon correlation - PARALLEL
*)
(* about 7 sec/546031 photons
*)
(* .085 sec / 10k pts
*)
(*****
*)
ClearAll[endCorrCompile];
Needs["CompiledFunctionTools`"];
endCorrCompile=Compile[
  {{a,_Integer,1},
  {b,_Integer,1},
  {wa,_Real,1},
  {wb,_Real,1},
  {tau,_Integer}},

```

```

Module[{bShift,g,jStart,jaMax,jbMax,
ja,jb,aa,bb,iPass,maxPass,wbb,q,done,endWhile2,sa,sb,waa},
  jaMax=Length[a];
  jbMax=Length[b];
  g=0.0;

  ja=0;
  jb=1;
  bShift=b+tau;

  Label[WhileStart1];
  bb=bShift[[jb]];
  wbb=wb[[jb]];

  Label[WhileStart2];
  q=(ja>=jaMax);
  If[q,Goto[done]];
  ja++;
  q=(aa= a[[ja]])>= bb;
  If[!q,Goto[WhileStart2]];
  q=(aa== bb);
  (*Print[ja," ", wa[[ja]]," ",wbb," ",g];*)
  If[q,g=g+wa[[ja]]*wbb];

  aa=a[[ja]];
  waa=wa[[ja]];

  Label[WhileStart3];
  q=(jb>=jbMax);
  If[q,Goto[done]];
  jb++;
  q=(bb=bShift[[jb]])>=aa;
  If[!q,Goto[WhileStart3]];
  q=(aa== bb);
  If[q, g=g+waa*wb[[jb]]];

  Goto[WhileStart1];

  Label[done];
  Return[N[g]]
],
RuntimeAttributes->{Listable},Parallelization->True(*,
RuntimeOptions["ErrorHandler"<Function[Throw[$Failed]]]*
];

```

```

(*CompilePrint[endCorrCompile]*)
(*xx={1,10,20,30,32,33,40,50};
www={1,1,1,1,1,1,1,1};
endCorr[xxx,xxx,www,www,2]//AbsoluteTiming;
endCorrCompile[xxx,xxx,www,www,2]//AbsoluteTiming*)

(*****
*)
(* enderlein single-photon SUMS
*)
(*****
*)
ClearAll[endSums];
endSums[a_,b_,wa_,wb_,tau_]:=
Module[{bShift,g,jStart,jMax,
ja,jb,jaMax,jbMax,aa,bb,iPass,maxPass,wbb,bLast,sa,sb,a1,wa
a,len,tMax,tMin,totLen},
(*Print["entering endCorr...."];*)
jaMax=Length[a];
jbMax=Length[b];
(*Print ["jaMax=",jaMax," ", "jbMax=",jbMax];*)
g=0;

If [jaMax!=0 && jbMax==0, (* counting up "a" *)
ja=0;
a1=a[[1]];
While[True,
If[ja<jaMax,ja++(*;Print["ja=",ja]*),Goto[done]];
If[( a[[ja]]-a1)>= tau,g=g+wa[[ja]]]
] (* end while*)
];(* end 'if' *)

If [jaMax==0 && jbMax!=0,(* counting up "b" *)
(*Print["entering '1b'...."];*)
jb=0;
bLast=b[[jbMax]];
While[True,
If[jb<jbMax,jb++(*;Print["jb=",jb,"
",b[[jb]]*),Goto[done]];
If[(bLast-b[[jb]])>= tau,g=g+wb[[jb]]]
] (* end while*)
]; (* end 'if' *)

If [jaMax==0 && jbMax==0, (* counting 1's *)
(*Print["entering '11'...."];*)
(* here wa, wb are actually a, b to provide times *)

```

```

tMax=Max[wa[[Length[wa]]], wb[[Length[wb]]]];
tMin=Min[wa[[1]], wb[[1]]];
totLen=tMax-tMin+1;
g=totLen-tau;
]; (* end 'if' *)

Label[done];
Return[g];
(*Print[g]*)
];(* end module 'endSums' *)
(*Print["\na=",a={1,10,20,30,32,33,40,50}]];
Print["b=",b={1,10,20,30,32,33,40,50}]];
Print["a=",a={1,2,3,4,5,6,7,8}]];
Print["b=",b={1,2,3,4,5,6,7,8}]];
Print["wa=",wa={11,1,1,1,1,1,1,111}]];
Print["wb=",wb={22,1,1,1,1,1,1,222}]];
Print["tau=",tau=6]];
Print["g=",g=endSums[a,a,wa,wa,tau]];
Print["g=",g=endSums[a,1,wa,wb,tau]];
Print["g=",g=endSums[1,b,wa,wb,tau]];*)

(*****
*)
(* enderlein sums - COMPILED
*)
(*****
*)
ClearAll[endSumsCompile];
Needs["CompiledFunctionTools`"];
endSumsCompile=Compile[
  {{a,_Integer,1},
   {b,_Integer,1},
   {wa,_Real,1},
   {wb,_Real,1},
   {tau,_Integer}},
  Module[{g,jaMax, jbMax,
ja,jb,tMax,tMin,totLen,q,iOrderEff,a1,aCrit,bCrit,bLast},
  jaMax=Length[a];
  jbMax=Length[b];
  g=0.0;
  ja=0;
  jb=0;

  q=(jaMax!=0 && jbMax==0);
  If[!q,Goto[endIf1]];
  aCrit=a[[1]]+tau;

```

```

ja=jaMax+1;
Label[whileStart1];
ja--;
q=(ja<1)V(a[[ja]]< aCrit);
If[q, Goto[done] ];
g=g+wa[[ja]];
Goto[whileStart1];
Label[endIf1];

q=(jaMax== 0 && jbMax!= 0);
If[!q, Goto[endIf2] ];
bCrit=b[[jbMax]]-tau;
jb=0;
Label[whileStart2];
jb++;
q=(jb>jbMax)V(b[[jb]]> bCrit );
If[q, Goto[done] ];
g=g+wb[[jb]];
Goto[whileStart2];
Label[endIf2];

Label[done];
Return[N[g]]
], (* end module 'endSumsCompile' *)
RuntimeAttributes->{Listable}, Parallelization->True];
(*CompilePrint[endSumsCompile]*)
(*Print["\na=", a={1,10,20,30,32,33,40,50}];
Print["b=", b={1,10,20,30,32,33,40,50}];
Print["a=", a={1,2,3,4,5,6,7,8}];
Print["b=", b={1,2,3,4,5,6,7,8}];
Print["wa=", wa={11,1,1,1,1,1,1,111}];
Print["wb=", wb={22,1,1,1,1,1,1,222}];
Print["tau=", tau=6];
Print["g=", g=endSumsCompile[a, a, wa, wa, tau]];
Print["g=", g=endSumsCompile[a, {}, wa, wb, tau]];
Print["g=", g=endSumsCompile[{}, b, wa, wb, tau]];*)

(*****
*)
(* enderlein count *)
(*****
*)
ClearAll[endCount];
endCount[a_, b_, wa_, wb_, tau_] :=
Module[{g, jaMax, jbMax, ja, jb, tMax, tMin, totLen, q},
jaMax=Length[a];
jbMax=Length[b];

```

```

tMax=Max[a[[jaMax]],b[[jbMax]]];
tMin=Min[a[[1]], b[[1]]];
totLen=tMax-tMin+1;
g=(totLen-tau) (**oneScaled*); (*Print["tot len=",totLen,"
",tMax, " ",tMin," tau=",tau];*)
(*Print[totLen , " ", tau, " ",g];*)
Return[N[g]]
)(* end module 'endSums' *)
(*Print["endCount=",endCount[a,b,wa,wb, (*tau*)0]];*)

(*****
*)
(* enderlein count - COMPILED
*)
(*****
*)
ClearAll[endCountCompile];
Needs["CompiledFunctionTools`"];
endCountCompile=Compile[
  {{a,_Integer,1},
  {b,_Integer,1},
  {wa,_Real,1},
  {wb,_Real,1},
  {tau,_Integer}},
  Module[{g,jaMax, jbMax, ja,jb,tMax,tMin,totLen,q},
    jaMax=Length[a];
    jbMax=Length[b];

    tMax=Max[a[[jaMax]],b[[jbMax]]];
    tMin=Min[a[[1]], b[[1]]];
    totLen=tMax-tMin+1;
    g=(totLen-tau) (**oneScaled*); (*Print["tot
len=",totLen," ",tMax, " ",tMin," tau=",tau];*)
    Return[N[g]]
  ],(* end module 'endSumsCompile' *)
  RuntimeAttributes->{Listable},Parallelization->True];
(*CompilePrint[endCountCompile]*)
(*Print["endCountCompile=",endCountCompile[a,b,wa,wb, (*tau*
)0]];*)

(*****
*)
(* deflate data by desired factor
*)
(*****
*)
ClearAll[deflateSub];

```

```

deflateSub[aInit_,waInit_,order_]:=Module[{qa,laNew,aa,wa},
  (*Print["starting deflations(s)"];*)
  qa=deflateCompile[aInit,N[waInit],order];
  laNew=If[order>0,Length[qa[[1]]]-1,Length[qa[[1]]]]; (*
drop incomplete last point*)
  laNew=Length[qa[[1]]]-1;
  aa=qa[[1]][[1;;laNew]];
  wa=qa[[2]][[1;;laNew]];
  Return[{aa,wa}];
  (*Print["deflations(s) done"];*)
  ];

(*****
*)
(* calc corr time (tau) for desired iorder, iPt
*)
(* NOTE! value returned in pre-deflated pts i.e. iOrder 0
pts      *)
(*****
*)
ClearAll[makeTau];
makeTau[iPt_, iOrder_,nPtsPerOrder_]:=Module[{index,tau,
tauCorr,iEffOrder},
  index=(iPt-1)+iOrder*nPtsPerOrder;
  tau=((2iOrder- 1)*nPtsPerOrder+(iPt-1)*2iOrder) (* enderlein
formula for tau values *);
  Return[tau]
  (*tauCorr=IntegerPart[tau*2-iEffOrder]*)
  ]; (* end module*)
(*Print["tau=",makeTau[2,1,10]];*)

(*****
*)
(* calc tot sum squares for gdd      *)
(*
*)
(*****
*)
ClearAll[gddSq];
gddSq[v1_,h1_,v2_,h2_]:= (v2-h2)2 (v1+h1) + (v1-h1)2 (v2
+h2) + v1 v2 + h1 h2 +v1 h2+v2 h1;

(*****
*)
(* attempt full time-tagged correlation
*)

```

```

(*****
*)
ClearAll[processPhotons];
processPhotons[a_,b_,waInit_,wbInit_,ra_,rb_] := Module[{(*maxOrder,nPtsPerOrder,l,aa,bb,qa,qb,wa,wb,outFileName,index,tauUsec,psPerPt,orderFactor,file,lNew,max,la,lb,laNew,lbNew,maxA,maxB,maxT,gList,sa,sb,tauList,nPts,minA,minB,nPtsA,nPtsB,totLen,(*span,*)done,gvv,gvh,ghv,ghh,gv1,glv,gh1,glh,g11,

tgvv,tgvh,tghv,tghh,tgv1,tglv,tgh1,tglh,tg11,tTauUsec,iOrder,msPerPt,

tauUsecTab,gvvTab,gvhTab,ghvTab,ghhTab,gv1Tab,glvTab,gh1Tab,glhTab,g11Tab,

tauCorrTab,corrFactTab,orderFactor2,orderFactor1,orderFactor0,maxWt*)},
  Print["starting 'processPhotons'"];
  SetDirectory[NotebookDirectory[]];

  (* inputs *)
  (* a, b = raw arr times ch a, b hvs *)
  (* wa, wb = num hv each arr time *)
  (* ra, rb = hv rate per DPC230 ch *)

  (* define constants *)
  maxOrder=46; (* ~11,259 sec = 2+ hr *)
  nPtsPerOrder=8;
  psPerPt=164.61; (*DPC230 psPerPt=164.61;*)
  usPerPt=psPerPt*10^-6;
  nPreDeflate=10; (* start with .16856 us/pt =2^10 *)
  (*Print ["ra=",ra, " rb=",rb];*)

  (*create blank lists *)
  tauUsecTab={};tTauUsec={};
  gv1v2Tab={};tgvv={};
  gv1h2Tab={};tgvh={};
  gh1v2Tab={};tghv={};
  gh1h2Tab={};tghh={};
  gh1Tab={};tgh1={};
  glhTab={};tglh={};
  gv1Tab={};tgv1={};
  glvTab={};tglv={};
  g11Tab={};tg11={};
  iOrderTab = {};tiOrder={};
  iPtTab={}; tiPt={};

```

```

lenTab={}; tLen={};
gddSqTab={};tgddSq={};
tauiOrderTab={};tTuiOrder={};
scaleTab ={};tScale={};

(* pre-deflate raw data *)
Print ["before predeflate:
Total[waInit]=" ,Total[waInit], "
Total[wbInit]=" ,Total[wbInit]];
qa=deflateSub(*uses
compile*) [a,waInit,nPreDeflate];qb=deflateSub(*uses
compile*) [b,wbInit,nPreDeflate];
aa=qa[[1]];bb=qb[[1]];
wa=qa[[2]];wb=qb[[2]];
laNew=Length[aa];lbNew=Length[bb];
Print ["after predeflate: Total[wa]=" ,Total[wa], "
Total[wb]=" ,Total[wb]];

(* NOTE! after this all times, etc. are in units of
2^(nPreDeflate) orig pts *)

(* work through desired orders *)
For[iOrder=0,iOrder<=1*maxOrder, iOrder++,

iTotOrder=nPreDeflate+iOrder;
s2a=ra*psPerPt*10-12*2iTotOrder; (*hv per this iOrder Pt
*)
s2b=rb*psPerPt*10-12*2iTotOrder;
(*Print ["s2a=" ,s2a, " s2b=" ,s2b];*)

(* deflate to desired order *)
If [iOrder>0,
qa=deflateSub(*uses
compile*) [aa,wa,1];qb=deflateSub(*uses compile*) [bb,wb,1];
aa=qa[[1]];bb=qb[[1]];
wa=qa[[2]];wb=qb[[2]];
laNew=Length[aa];lbNew=Length[bb];
];

orderFactor2=2.-iOrder;
orderFactor1=2.0;
orderFactor0=2.iOrder;

span=Max[aa[[laNew]],bb[[lbNew]]]-
Min[aa[[1]],bb[[1]]];
totLen=span+1;
(*If[iOrder==17,

```

```
iOrderSave=iOrder;aaSave=aa;bbSave=bb;waSave=wa;wbSave=wb];
*)
```

```
(* make tables of tau's and flags for parallel calc of
correlations for given iOrder*)
```

```
tauOrderTab=Table[IntegerPart[makeTau[iPt,iOrder,nPtsPerOrder]
*2-iOrder],{iPt,nPtsPerOrder}];
tauUsecTab=tauOrderTab*2iTotOrder*psPerPt*1^-6;
corrFactTab=Table[If[totLen-tauOrderTab[[iPt]]>0,N[totLen/(totLen-
tauOrderTab[[iPt]])],0],{iPt,nPtsPerOrder}];
ok=Table[tauOrderTab[[i]]<=span,{i,nPtsPerOrder}];
If [!ok[[1]],Break[]];
```

```
(*gvvTab=Table[If[ok[[iPt]],endCorrCompile[aa,aa,wa,wa,tauOrderTab[[iPt]]],0],{iPt,nPtsPerOrder}];
gvhTab=Table[If[ok[[iPt]],endCorrCompile[aa,bb,wa,wb,tauOrderTab[[iPt]]],0],{iPt,nPtsPerOrder}];ghvTab=Table[If[ok[[iPt]],endCorrCompile[bb,aa,wb,wa,tauOrderTab[[iPt]]],0],{iPt,nPtsPerOrder}];ghhTab=Table[If[ok[[iPt]],endCorrCompile[bb,bb,wb,wb,tauOrderTab[[iPt]]],0],{iPt,nPtsPerOrder}];
gv1Tab=Table[If[ok[[iPt]],endSumsCompile[aa,{},wa,{},tauOrderTab[[iPt]]],0],{iPt,nPtsPerOrder}];glvTab=Table[If[ok[[iPt]],endSumsCompile[{},aa,{},wa,tauOrderTab[[iPt]]],0],{iPt,nPtsPerOrder}];gh1Tab=Table[If[ok[[iPt]],endSumsCompile[bb,{},wb,{},tauOrderTab[[iPt]]],0],{iPt,nPtsPerOrder}];glhTab=Table[If[ok[[iPt]],endSumsCompile[{},bb,{},wb,tauOrderTab[[iPt]]],0],{iPt,nPtsPerOrder}];
g11Tab=Table[If[ok[[iPt]],endCountCompile[aa,aa,wa,wa,tauOrderTab[[iPt]]],0],{iPt,nPtsPerOrder}];*)
```

```
(* parallel calc of correlations for given iOrder w/
Compile*)
```

```
gvvTab=ParallelTable[If[ok[[iPt]],endCorrCompile[aa,aa,wa,wa,tauOrderTab[[iPt]]],0],{iPt,nPtsPerOrder}];
```

```
gvhTab=ParallelTable[If[ok[[iPt]],endCorrCompile[aa,bb,wa,wb,tauOrderTab[[iPt]]],0],{iPt,nPtsPerOrder}];ghvTab=ParallelTable[If[ok[[iPt]],endCorrCompile[bb,aa,wb,wa,tauOrderTab[[iPt]]],0],{iPt,nPtsPerOrder}];ghhTab=ParallelTable[If[ok[[iPt]],endCorrCompile[bb,bb,wb,wb,tauOrderTab[[iPt]]],0],{iPt,nPtsPerOrder}];
```

```

tauiOrderTab[[iPt]],0},{iPt,nPtsPerOrder}];

gv1Tab=ParallelTable[If[ok[[iPt]],endSumsCompile[aa,{},wa,{}],
],
tauiOrderTab[[iPt]],0},{iPt,nPtsPerOrder}];g1vTab=Parallel
Table[If[ok[[iPt]],endSumsCompile[{},aa,{},wa,{}],
tauiOrderTab[[iPt]],0},{iPt,nPtsPerOrder}];gh1Tab=Parallel
Table[If[ok[[iPt]],endSumsCompile[bb,{},wb,{}],
tauiOrderTab[[iPt]],0},{iPt,nPtsPerOrder}];g1hTab=Parallel
Table[If[ok[[iPt]],endSumsCompile[{},bb,{},wb,{}],
tauiOrderTab[[iPt]],0},{iPt,nPtsPerOrder}];
g11Tab=Table[If[ok[[iPt]],endCountCompile[aa,aa,wa,wa,
tauiOrderTab[[iPt]],0},{iPt,nPtsPerOrder}];
scaleTab=Table[2^iOrder,{iPt,nPtsPerOrder}];
iOrderTab=Table[iOrder,{iPt,nPtsPerOrder}];
iPtTab=Table[iPt,{iPt,nPtsPerOrder}];
lenTab=Table[totLen,{iPt,nPtsPerOrder}];

(*gv1v1v2v2Tab=ParallelTable[If[ok[[iPt]],endCorrCompile[aa
,aa,wa^2,wb^2,tauCorrTab[[iPt]]],0},{iPt,nPtsPerOrder}];*)

(* append results this iOrder to those of previous
iOrder's *)
tTauUsec=Join[tTauUsec,tauUsecTab];
tgvv=Join[tgvv,gvvTab*corrFactTab*orderFactor2];
tgvh=Join[tgvh,gvhTab*corrFactTab*orderFactor2];
tghv=Join[tghv,ghvTab*corrFactTab*orderFactor2];
tghh=Join[tghh,ghhTab*corrFactTab*orderFactor2];
tgv1=Join[tgv1,gv1Tab*corrFactTab*orderFactor1];
tg1v=Join[tg1v,g1vTab*corrFactTab*orderFactor1];
tgh1=Join[tgh1,gh1Tab*corrFactTab*orderFactor1];
tg1h=Join[tg1h,g1hTab*corrFactTab*orderFactor1];
tg11=Join[tg11,g11Tab*corrFactTab*orderFactor0];
tScale=Join[tScale,scaleTab];
tiOrder=Join[tiOrder,iOrderTab];
tiPt=Join[tiPt,iPtTab];
tLen=Join[tLen,lenTab];
tTauiOrder=Join[tTauiOrder,tauiOrderTab];

];(* end for 'iOrder' *)

Return[{tTauUsec,tgvv,tgvh,tghv,tghh,tgv1,tg1v,tgh1,tg1h,tg
11,tScale,tiOrder,tiPt,tLen,tTauiOrder}];
];(* end module - 16 output vars *)

```

```

(*****
*)
(* full processing of 'outRaw.csv' file
*)
(*****
*)
ClearSystemCache[];
ClearAll[];
SetDirectory[NotebookDirectory[]];
Off[CompiledFunction::cfta];

(*out=makeChannelArrays["hvSimOutRaw.csv"];*)
(*{t1,t2,l1,l2,w1,w2,gTradEst,maxTimeSec, r1,r2, gGbEst}*)
out=makeChannelArrays["OutRaw-2015-04-08-IgE-cell 10-dot1-
1.csv"];
t1=out[[1]];
t2=out[[2]];
tm=out[[3]];
l1=out[[4]];
l2=out[[5]];
w1=out[[6]];
w2=out[[7]]; (*{t1,t2,l1,l2,w1,w2,gTradEst,maxTimeSec,
r1,r2, gGbEst}*)
wm=out[[8]];
latestPt=out[[9]];
maxTimeSec=out[[10]];
r1=out[[11]]; (*hvPerSec*)
r2=out[[12]]; (*hvPerSec*)
gTradEst=out[[13]];
gGbEst=out[[14]];
(*n=10000;
t1={};
t2={};
w1={};
w2={};
For [i=1,i≤ n,i++,
x=RandomVariate[PoissonDistribution[.1]];
y=RandomVariate[PoissonDistribution[.1]];
(*x=i;*)
AppendTo[t1,i];
AppendTo[t2,i];
AppendTo[w1,x];
AppendTo[w2,y]
];
Print["tot w1=",Total[w1]];
Print["tot w2=",Total[w2]];*)

```

```

Monitor[q=processPhotons[t1,t2,w1,w2,r1,r2],{"iOrder=
"iOrder}];//AbsoluteTiming
ClearAll[outData,outDataRect];
outData=Append[q,{"",l1=l2="time=",r1=r2="gTrad
Est","gGbEst="}];
AppendTo[outData,{"",l1,l2,maxTimeSec, r1,r2,
gTradEst,gGbEst}];
outDataRect=Transpose[PadRight[outData]];
headings={"uSec","gvv","gvh","ghv","ghh","gv1","g1v","gh1",
"glh","gl1","scale","iOrder","iPt","len","tauiOrder","param
","val"};
outDataRect=Prepend[outDataRect,headings];
If[FileExistsQ["gg.csv"],DeleteFile["gg.csv"]];
Export["gg.csv",outDataRect];
1.12953*106 494492.
len tot= 1624017
len 1= 1129525
len 2= 494492
w 1= 1.12953*106
w 2= 494492.
maxInitPts= maxInitPts$2365
max Time= 30.0002
r1= 37650.6
r2= 16483.
gTradEst= 2.28421
gGbEst= 0.391026
starting 'processPhotons'
before predeflate: Total[waInit]= 1.12953*106
Total[wbInit]= 494492.
after predeflate: Total[wa]= 1.12952*106 Total[wb]=
494491.
{49.2014,Null}

```

List of Abbreviations

APD	Avalanche Photodiode Detector
BFP	Back Focal Plane
CFDA	Continuous Fluorescence Depletion Anisotropy
CSV	File-comma separated values file
CTAB	Cetyltrimethylammonium bromide
DNP-BSA	Dinitrophenyl bovine serum albumin
EM	Electron microscopy
FcεRI	Type I Fcε receptor
FDA	Fluorescence Depletion Anisotropy
FRAP	Fluorescence Recovery after Photobleaching
FPR	Fluorescence Photobleaching Recovery
GPCR	G-protein-coupled receptor
IgE	Immunoglobulin E
LH receptor	Luteinizing hormone receptor
MβCD	Methyl-beta-cyclodextrin
NA	Numerical Aperture
NB file	Mathematica Notebook
PFD	Polarized Fluorescence Depletion
QD	Quantum dot
RCT	Rotational correlation time
ROI	Region of interest

TACF	Time-autocorrelation function
TPA	Time-resolved phosphorescence anisotropy
TTSPC	Time-tagged single photon counting
TTTR	Time-tagged time-resolved